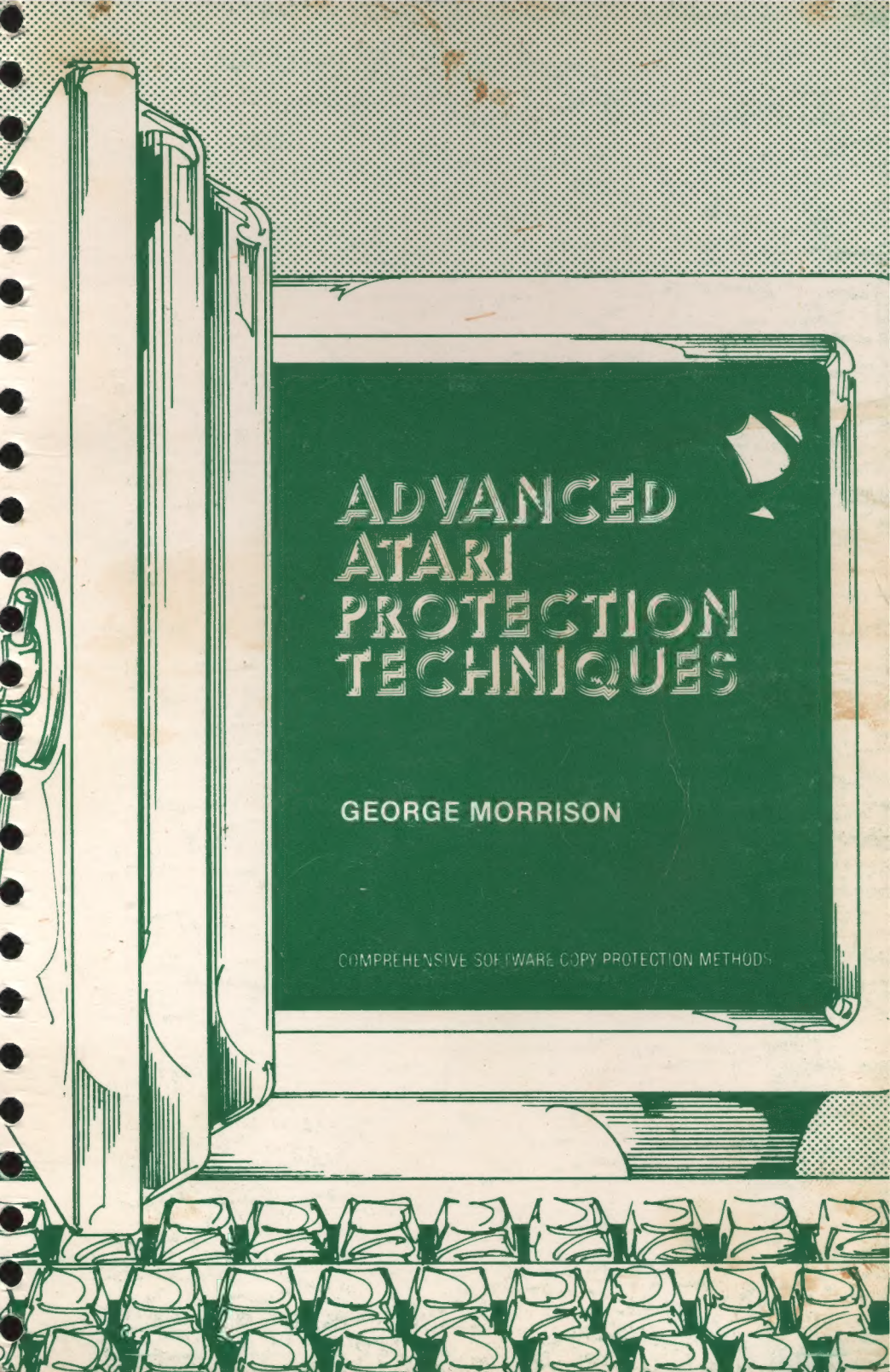

ADVANCED
ATARI
PROTECTION
TECHNIQUES
GEORGE MORRISON
COMPREHENSIVE SOFTWARE COPY PROTECTION METHODS

Alpha Systems Presents

ADVANCED ATARI PROTECTION TECHNIQUES

Volume II of the Software Protection Series

By George Morrison
Contributors: Helen Prozialeck
George Polly
John Liang

Art and Cover Design: Lee Kirshbaum

1st Printing July, 1986

Manufactured in the United States of America

10 9 8 7 6 5 4 3

ACKNOWLEDGEMENTS

I would first like to thank those who have patiently waited while this book was being produced. The difficulties in keeping a state-of-the-art book up to date have led to several delays, but I hope you will agree that the result is well worth it. I would also like to acknowledge the people without whom this book could not have been written; Helen Prozialeck, Debbie Muster, George Polly, Lee Kirshbaum, Ethel Morrison, Bonnie Lawrence, Craig Walters, Al Wylcznski, Jeff Bader, and Craig Wolf.

PREFACE

Whenever the subject of home computers comes up, the topic of piracy inevitably follows. The two are, to some people, inseparable. Piracy has both fierce enemies and staunch defenders. The emotions surrounding piracy are so strong that rational arguments are often obliterated by personal feelings.

Piracy is, indeed, a serious problem facing software manufacturers who cater to the home computer market. On the other side of the coin, not every user is a pirate. Software publishers have a delicate balancing act to perform, they must protect their software from pirates, with a minimum of inconvenience to legitimate users.

This book takes an honest look at all of the aspects surrounding piracy. It discusses the advanced methods of software protection manufactures have developed to thwart pirates. It also reviews some popular back-up tools, and examines the various kinds of pirates. The disk included with this package provides programs which will help anyone, from individual to large corporation, to keep their software safely out of pirate circles.

Helen Prozialeck

ADVANCED PROTECTION TECHNIQUES
TABLE OF CONTENTS

SECTION III
THE TOOLS

Introduction

Piracy is the illegal duplication and distribution of copyrighted software. In spite of laws prohibiting piracy, it has become one of the most popular pastimes among computer users. Although users have a legitimate right to make backups for personal safekeeping, thousands of dollars in software sales will be lost each month due to piracy. The plummeting cost of personal computers has been one of the major contributing factors to the growth of piracy. When a complete computer system costs only several hundred dollars, most users are understandably reluctant to pay $100 for a single program, especially if he or she can make a copy of a friend's program for the cost of a blank disk. Both software companies who demand outrageous prices, and users who expect something for nothing are to blame for this dilemma.

Software companies are caught in a bind. Software development, testing, and marketing can be prohibitively expensive. Most users actively dislike copy protected software, but unprotected software, even buggy, preliminary versions, are rapidly spread through pirate circles.

And the software does spread fast. If a single unprotected copy of an unreleased program gets into a pirate's hands, the market for that product can be completely ruined in a matter of weeks.

Software spreads this quickly because pirates are not only numerous and widespread, but very organized. Each pirate is anxious to get the newest programs, and will quickly trade his latest acquisitions for the next 'hot' program. The

telephone system has rendered distances meaningless. With electronic means, a pirate can transmit a program across the country as quickly as he can transmit it across town. Pirates form groups, clubs, bulletin boards, parties, and newsletters to facilitate the trading of illegally copied software.

Some companies have recognized the size of the pirate market, and produced materials that cater to it rather than thwart it. Companies advertise hardware products designed to produce back up copies of copy protected software. Software companies produce tools designed to make a pirate's task much easier. As some software producers toil to protect their programs, others offer utilities and hardware modifications to break and copy even the most complex protection schemes.

Some software companies have chosen the legal route to software protection, pushing legislatures to toughen laws, and law enforcement officials to carry them out. Police have set up bogus bulletin boards to lure unsuspecting pirates. Government officials in both the United States and Canada have begun to crack down on profiteers, the pirates who reproduce copyrighted software in large quantities, and offer them for resale at bargain prices. These unscrupulous pirates often convince unwitting buyers that they are purchasing the 'real thing' instead of a bootleg copy of a stolen program.

This book will delve into pirates themselves, who they are, and how and why they copy programs. It will explain the copy protection techniques used to fight piracy today, and the ones that many companies are developing to use tomorrow. It will examine the current status of copyrights, patents, and other legal forms of copy protection, discuss the pros and cons of each, and analyze the trends in software protection law. It will review the various miscellaneous methods used by software producers to protect their goods. It will review the current off the shelf software back up tools, discussing the capabilities, advantages, and disadvantages of each. Lastly, this book and disk offers some utility programs for software writers who want to guard their programs from pirates.

For the most recent news, too late to be printed here, see the disk included with this package.

Important Warning to Software Manufacturers:

Software protection methods often take advantage of flukes in the way a computer or disk drive perceives data. With the increasing number of models of computers and drives, it is difficult to be sure that the fluke will behave the same in each. One of the greatest frustrations a purchaser can experience is to find that a legitimately purchased program will not run on his system because of copy protection. Manufacturers should be sure to thouroughly test all protection methods on a number of configurations to be sure it works properly on each. If this step is overlooked, it can create bad will towards the product (and the company) that may be difficult to overcome. If you find that your protection does not work with all brands of disk drive, computer, etc., you should clearly mark that information on the package, or, better yet, change the protection.

SECTION I
THE PEOPLE

INTRODUCTION

Computer use today spreads across all walks of life. The growing popularity of personal computers has brought news about computer hackers to the front pages of publications around the country. Terms such as Phone Phreaks and Computer Pirates are popping up in the news every day. Most news writers, and most people, are not aware of what is really happening in these well publicized cases, and do not really understand the underlying trends. Understanding these trends is an essential first step to understanding the trends in software protection. This section of the book will try to clear up the confusion, examine the cases, and show the ways to deal with the people involved. It will focus primarily on Piracy and On-Line Security, but also cover the related areas of Phreaking.

This section will break the people down into groups and talk about each group separately, then show how they relate to the whole topic. In several areas such as Phreaking and Pirate Boards, the technical details of how they operate will be dealt with. These areas are included for the more advanced readers, but full understanding is not required.

"They call us pirates and worse. They lock up their programs behind hardware and software schemes. They set the minions of the law upon us. And still we flourish by our wiles.

Ahoy, ye microlubbers: to pirate a program is not to steal, but to liberate knowledge. We don't take money or goods from anyone; we merely free up information. Most of us don't profit from our buccaneering activities; instead, we share the wealth with our fellow computer users.

The software moguls have only themselves to blame for our cracking open the bars to their programs. If they didn't charge a king's ransom for disks that cost a pittance to duplicate, there would be little incentive for us to practice our skills. There would be no need for them to protect their programs if software were no more expensive than what you and I can afford to pay.

We are no longer in the Dark Ages of personal software, when so few people used computers that program development costs had to be defrayed by high unit prices. Now so many microcomputers are in use that a program should cost no more than a lightweight paperback novel. Instead, we are paying illuminated manuscript prices.

Maybe someday the software publishers will understand how they're killing off the golden goose. But until that time be warned: there will be many a pirate's flag on the software horizon."
JOLLY ROGER

From Digital Deli by Gerry J. Elman

Chapter 1

MOTIVATIONS

Piracy is the duplication and distribution of copyrighted software without the permission of the copyright owners. It is perfectly legal to make a working back-up copy of software that you have purchased, as long as that back-up copy is only for your own personal use. The problems arise when back-up copies are given away to others who have not purchased the software, or traded for copies of other software that the individual does not own or intend to buy. Many people who are unsure or misinformed about copyright laws are unintentional pirates. In a survey by Allen Harberg of 100 Atari user group presidents, up to 62% believed that certain activities, which do, in fact, violate copyright laws, were legal!

The reasons for becoming an intentional pirate are as diverse as computer users themselves. Generally speaking, there are two main kinds of pirates; those who pirate for personal enjoyment, personal satisfaction, or financial savings, and those who pirate for profit. The people who pirate for enjoyment can be loosely broken down into the categories of hackers, collectors, gamers/users, and belongers.

Hackers

A Hacker is a person with a strong personal interest in computers. True Hackers are very closely

involved with their computers and the software available for them. They can program, and frequently they are very talented programmers. They have an excellent understanding of their systems. Hackers are extremely curious, and usually see software protection as a puzzle to be solved. They don't just stop at removing the protection from a program, but will modify the program to suit themselves, often adding an "unlimited life" option to a game, or encrypting their initials or "handle" into the title screen. These people have the drive and skill to disarm most software protection schemes. Fortunately for software producers, true hackers are a small minority of computer users (more about hackers in Chapter 4 under Phone Phreaks).

Collectors

The collector is, in some ways the most dangerous kind of pirate. Collectors want to build a huge collection of software, just as some people build huge coin or stamp collections. Collectors will make several pirate copies of programs, and use these copies as trading material to obtain more illegal copies of other software. Some Collectors will go to extremes to get a copy of a new or heavily advertised program, even though they have little desire to use it. While a hacker knows every detail of his programs, a collector may have run them only once to verify that they work. He may never use a program again, except to trade it for other software. This group has the least knowledge of their software, and is the most likely group to be seen trading preliminary versions.

Gamers/Users

The majority of computer users who are involved with piracy are gamers and users. They accumulate and trade software because they want to use one or more specific programs. The motivation to pirate is based mainly on saving money. Most often this group is made up of people who love to play games. Most people use only one good

spreadsheet or word processor program, but gamers want to play as many different good games as possible, so most of the programs they trade are game software. When a new game comes out, these people will often band together to buy it and hope to make copies for each other.

Belongers

Belongers trade software to be part of a group. They like to meet people, and have found the computer is a good way to do it. Belongers have an extensive software collection to gain status among other computer users, or trade software with a large number of people to be sociable. Although the people who use pirated programs to gain status are usually in their teens, these traits can be found all the way to the oldest computer users as well.

Profiteers

A wide range of activities can fall into this category. On one end of the spectrum are the commercial ventures that make counterfeit copies of programs to sell to unsuspecting purchasers, and on the other is a teenager who trades his software for blank disks to expand his collection. The commercial pirates make up the majority of software companies in countries like Taiwan, but are not very active in the U.S. This is partially due to the lack of copyright agreements in some countries as well as the general aura of acceptability surrounding piracy there. Between these two extremes lies a group who may sell pirated software, but only to friends, and often under the pretense of covering expenses incurred in obtaining the software. It should be pointed out that the majority of pirates are happy to make copies for themselves or their friends, but draw a definite line before selling any pirated software. Some will buy a program, copy it, and sell the original at a discounted price.

There has been a recent rash of FBI sting operations to close down blatantly obvious profiteers who sell counterfeit software through ads in newspapers and computer magazines (see New Trends in Software Law).

Conclusion

One fact not mentioned in the above discussion is that piracy is illegal. A copyright notice on software seems to have very little effect on the majority of software pirates. A company that feels a copyright notice is adequate protection against piracy is ignoring the facts. Rumors of persons being jailed for copyright infringement have scared some people, but most people know these rumors are not true (see New Trends in Software Law section).

Most people are a mix of personality types, and most pirates will fit in more than one category. Still, understanding what motivates a pirate can provide clues to successful software protection.

Chapter 2

THE SOURCES

Where does it all begin? Where are the sources that supply pirates with their goods? How is it that pirates often have products well before they are available in stores, sometimes even before they have been announced to the public? There are a variety of answers to these questions. This section discuss some of the most common sources.

Purchasers

Purchasing is the most obvious source for software, but is probably the least used among computer pirates. They avoid this method for several reasons. First, it is the slowest, since special ordering a just released or newly announced product can take a long time. New or unreleased software is the most highly prized among pirates, so most pirates are not interested in software that is already available in stores. Purchasing is risky for a pirate, because he may be registered with the product's serial number at the time of purchase. If illegal copies bearing that serial number are found in circulation, the source is easily tracked down. It's the most expensive method. Lastly, purchased copies are the hardest to back up, since all the protection is intact.

A purchaser has some advantages. He will get a complete set of documentation, and company support if assistance is needed. He will receive a version without extra bugs accidentally added to the program by the person who broke the protection. These are crucial points when working with business

or productivity software.

Since these areas are of minimal importance for most game software, games are usually obtained from other sources.

Samples & Beta Versions

Surprisingly, many companies who seem concerned about piracy release totally unprotected copies, called beta copies, of their software for testing and samples. Some companies give out unprotected copies to reviewers to avoid any problems the protection might cause the reviewer, and to prevent the reviewer from mentioning the protection in his review. Judging from the number of unfinished programs, or programs labeled 'preliminary version' or 'for demonstration only, not to be sold', available in the pirate community, beta and test copies of programs are clearly getting into pirate's hands.

The old Atari management was probably the worst at this kind of offense. They had a program where unprotected game software was distributed to high school students for review. Naturally, this unreleased software was excellent trading material, and so spread very quickly. This practice destroyed the market for a good number of games created by Atari programmers or those companies (like Lucasfilm) who were marketing their software through Atari. The 'new' Atari Corp. seems to have solved most of these problems.

Software Company Insiders

Frequently, company employees are given preliminary copies of software for testing purposes. These copies have no protection and are usually kept as disk files until the final protected versions are ready, at which point the software is placed on a boot disk, cartridge, etc. Surprisingly often, these preliminary versions make their way into the hands of pirates. Since they are unprotected and can easily be transmitted over a modem, they spread around the country (even the world) very fast. In rare cases this can work to the companies advantage. If the

program is far from completion, but still looks very good, It can whet the appetite of potential buyers. Usually, however, the program is virtually complete and advance copies can destroy the market before the company even has a chance to make a sale.

Company leakes allow the pirate community to obtain many programs that are available through no other channels, especially programs that were never officially marketed. This actually entices many people to go into piracy.

Hardware Company Insiders

Computer manufacturers like to show potential buyers an abundant selection of low priced software for their machines. Sometimes this desire to show buyers how much quality software is available has led to piracy. Software companies often supply hardware vendors with programs (even before general release) for testing purposes and marketing assistance. Hardware company employees have been known to circulate pirate copies of both their own companies software, and test copies of software from third party vendors. Apple Computer, Inc. is a well known example of this problem. In a recent investigation, reporters from InfoWorld obtained over $1100 worth of illegally copied software from a single Apple employee, including a copy of an unreleased spreadsheet program from Microsoft, called Excel. None of the programs were copy protected. The employee admitted to frequently making pirated copies of programs, and trading with other Apple employees.

This kind of piracy has destriyed the market for some programs even before they were released. In 1984, Don Brown of CE Software sent pre-release copies of several software programs to a hardware manufacturer and another third party software developer. Within weeks, copies of the program appeared on bulletin boards around the country. They began to receive technical support questions well before the products were released. Brown said "It ruined the market for the product.", and he was forced to abandon attempts to market the product through retail outlets.

Although most hardware companies may take an official hard-line stance against piracy, few will actually do anything about it. It is best for them if a lot of software is easily available for their machines. Large quantities of cheap (or free) software is a good selling point. Some buyers have been known to select a particular brand of computer on the basis of access to pirated software.

Other Sources

Any time a pirate is exposed to an unprotected copy of a program, there is a chance the program will be stolen. Some of the most common places where a pirate can get a copyable program are listed below.

Computer Store Employees

A pirate working in a store has the opportunity to copy and distribute all the programs in the store (complete with documentation), as well as any samples, demos, and pre-release copies a store might receive.

Users Groups

Only a few users groups are actually pirate groups (see Distribution Methods). Users groups are often given pre-release software for evaluation and testing. If just one pirate gets an unprotected version of that software, it can quickly spread throughout the pirate community.

Libraries and Schools

Most software companies know that schools are frequently involved in piracy. Teachers will make copies of a program for entire classes, and the students will make copies of those copies. Some libraries have begun to carry software. A pirate will simply borrow the software and documentation, copy it, and return it.

Work Locations

Business software is usually purchased by companies for each legitimate user. However, employees have been known to make copies for personal use and trading material.

Shows

It's surprising how often software still months away from being completed is stolen or copied from displays at trade shows. Some vendors remove the disks after loading the software, and many shows restrict admission to those 18 and older, but shows continue to be a source for many advance copies and preliminary release versions traded in pirate circles.

Conclusion

With the large number of sources open to pirates it is not surprising that piracy is as widespread as it is. Although it is impossible to close off all a pirate's sources, software producers should try to reduce the risk in every area. One good way to discourage software company insiders from pirating is to give employees with access to unprotected software a percentage of the income generated from software sales. This way they have a personal interest in keeping the program away from pirates. That practice, in addition to careful employee screening and control procedures, will go a long way toward reducing the piracy problem.

When dealing with hardware companies, securing a non-disclosure agreement before delivering software is essential. Also, be sure the program code itself is clearly marked not for distribution.

Most importantly, never release an unprotected copy, unless you wish to forego copy protection.

Chapter 3

NON-ELECTRONIC DISTRIBUTION

Once pirates obtain software, they will copy and distribute it to others by various means. The techniques they use to make the copies vary depending on the type of software protection and documentation. The methods used to copy the software are covered in Section II of this book. The remainder of this section is devoted to the methods pirates use to distribute software.

Since software pirates are computer users, electronics play a major role in the distribution of pirated software, though some pirates find it quicker and more convenient to trade software by non-electronic means. Because piracy brings people from very diverse backrounds together, it is not unusual to find someone over 40 trading with a high school student. Pirates have created many activities to make trading software easier and faster. This section will describe some of these methods.

Individual Traders

The easiest and probably most widely used method of distribution is individual pirates trading among their friends and acquaintances. They get together and exchange their software while discussing new programs and demonstrating software. Each person may have a small group of people with whom they trade. Although software spreads slowly at first, it rapidly picks up speed as more and more people get it and trade it to others. Pirates spend hours trading, and will pass the software along as soon as they get it. Often it is exchanged only on the promise that it won't be spread any further. These promises travel along with the software as it is traded

from person to person, each person thinks it won't hurt to give the program to one more person as long as that person promises not to give it to any one else.

The Mail System

The mail system is an extension of individual trading in which software and documentation is mailed back and forth between pirates. Obviously, this greatly extends the reach of individual pirates allowing them to exchange software with people from all over the world. Fortunately for software producers, there is enough distrust among pirates that this method is seldom effective. Most pirates expect to receive other programs in exchange for copies of the software they own. They often hesitate to mail software to other pirates, because they have no assurances the other pirates will return the favor.

User Groups and Pirate Groups

Many people get involved with piracy through pirates they meet at users groups. Although most users groups officially discourage piracy, individual members can be seen trading copyrighted material during meetings. Sometimes user group's public domain libraries inadvertently contain copyrighted materials, and often even well-meaning user groups will duplicate articles and programs from magazines without permission.

So far the topic has focused on legitimate user groups, but many groups are formed with piracy as their major function. Pirate groups with names like the Pittsburgh Pirates and National Atari Pirates Organization (NAPO) can have over a hundred members and print newsletters describing the newest pirated software. They often publish lists of the software available to members and a 'Want List' of titles not yet in the library. Some have membership fees and dues that go towards maintaining a library of pirated disks, and making new purchases. Often the group will purchase special hardware like Happy Enhanced disk drives for use by group members. Some groups have well over 1,000 disks, each filled with copyrighted programs, in their library.

The justification members use are like those used by all pirates. One unique pirate group charter states that "The purpose of this group is for it's members to provide off-site back-up service to other members". Others say they meet to help offset the cost of overpriced software,

or even such lofty goals as to usher in a new era with free exchange of information, or to assist the 'informationaly deprived'.

The danger of pirate groups is that the distribution of software is very organized. By pooling together the talents and purchasing power of the members, they can back-up a large amount of software. Pirate groups also tend to encourage piracy by giving it an aura of social acceptability and making a large collection of pirated software a status symbol. A skilled hacker in a group like this receives the praise and encouragement of others, and the organized skills of the members means that software can easily be distributed.

Pirate Parties

When a pirate invites other pirates from near and far for a day of copying, they call it a pirate party. They bring as many systems and disk drives as possible and quickly and effectively exchange a lot of software. This brings together enough software at one place that people are willing to travel relatively long distances to participate. Some pirate users groups also use this method for exchanging software. It's used most in cases where special hardware is needed to make copies.

Documentation

So far, this software exchange seems easy. Just pop in a blank disk and off you go. But some software is worthless without the manuals on how to use it. Of course any Xerox machine can help, but many pirates refuse to let documentation out of their sight even for a short time. The difficulties involved in obtaining copies of documentation make it one of the major stumbling blocks for pirates to overcome.

Recognizing this, companies have tried to make their documentation even more needed. They can implement various passwords, make the program more difficult or complex, or use one of the other methods detailed in Chapter 10, Other Protection Methods. If not for the problems documentation poses to pirates, piracy would be even more widespread than it is today.

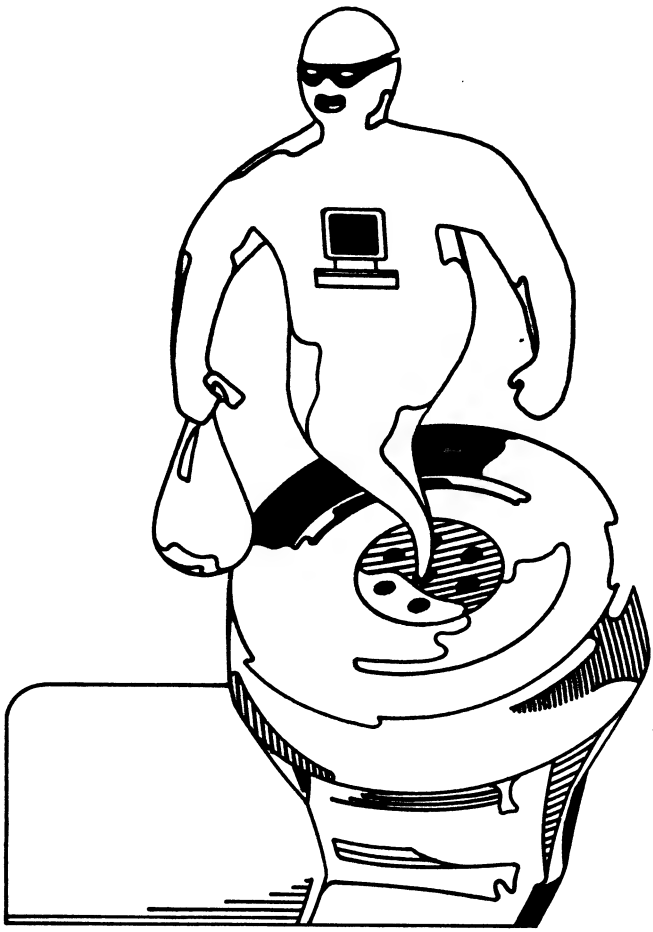
Chapter 4

ELECTRONIC DISTRIBUTION

Electronic distribution of software holds many advantages for a software pirate. First and foremost is speed. Software can be transmitted around the world in a matter of minutes. Next to speed is the ability to make distances a less significant obstacle. If line quality and long distance charges are not considered, software could be exchanged with a pirate in China as easily as it could go across the street. Another factor is ease. A pirate can put up an on-line bulletin board system and trade software with people from all over the country without even being at home. A user can wake up at 4 AM and download programs without getting dressed, or disturbing the person sending him the software.

"Many people think of phone phreaks as slime, out to rip off Bell for all she is worth. Nothing could be further from the truth! Granted, there are some who get their kicks by making free calls; however they are not true phone phreaks. Real phone phreaks are 'telecommunications hobbyists' who experiment, play with, and learn from the phone system. Occasionally this experimenting, and a need to communicate with other phreaks (without going broke), leads to free calls. The free calls are but a small subset of a >true< phone phreaks activities."

The Magician (Noted Phone Phreak)



PHONE PHREAKS & ILLEGAL USE OF PHONE LINES

A phone phreak is a person who uses a telephone in improper or illegal ways. "Phone Phreaking" is participating in these telephone related activities. Phone phreaking does not necessarily involve breaking into on-line systems. That activity is referred to as cracking. Cracking, hacking, and other activities closely related to phone phreaking are discussed in the next section. This section deals with phone use.

Phreaking is closely associated with computer use. Although a computer is not required, almost all phreakers today use a computer to take advantage of the telephone system's special circuitry. Some of the most common activities of a phone phreak are to "hack out" (use trial and error methods) telephone long distance access codes, trade access codes, and use access codes illegally to make long distance telephone calls, often to on-line bulletin boards. Most phreaking activities are performed for enjoyment or challenge rather than any kind of financial rewards. Most phone phreaks consider themselves as telecommunications hobbyists.

The phreaks who do get involved for financial reasons (free phone calls) are also the most likely to be involved with software piracy. A successful phone phreak is dedicated, resourceful and intelligent. A phone phreak who wants to copy a program will usually have no trouble breaking even the most sophisticated software protection scheme. A pirating phone phreak has the ability to transfer the broken copy easily and quickly at little or no cost. As a result, phreaking is often associated with software piracy.

Hacking (As Related to Phone Phreaks)

An enthusiastic phreak can quickly run up a big phone bill. To keep the costs down, one of the most popular activities among phreaks is 'hacking' out

the charges. Since the charges are billed to the loop number, the phreakers can talk for free.

Phone phreaks use boxes that generate tones to fool the phone company's computers. One popular trick is to call a toll free 800 number. When a call is placed to an 800 phone number, the local billing computer will not charge the caller. Then, when the 800 number is ringing, before anyone has a chance to answer the phone, the phreak uses a box to send a tone down the line. This makes the machine at the other end of the circuit think the caller has hung up before the call was answered. The caller's end of the circuit is still active, so the phreak can send tones through the line that will route his call anywhere in the world. When the receiving party finally picks up the phone, the call information will be sent back to the billing computer. The billing computer, however, still thinks the caller dialed an 800 number, so it throws the billing information away. The result is a free phone call.

Black Boxes, Blue Boxes, and Other Exotic Hardware

Boxes are the tools of the trade for phone phreaks. Although a computer can now do some of the processing done by these devices, these boxes are still popular tools for phreaks.

Boxes

BLACK - Also known as a "MUTE" box, this box causes the phone company's computers to think that a call was never answered, so it's never billed, but allows both parties to have a conversation. This was the first of the many phone phreak boxes. There's more information about black boxes in the following section.

BLUE - Gives the user the power of a long distance operator. It's very powerful for routing and directing calls. AT&T has found a way to detect these, so they are almost always used from pay phones.

know tricks to make it go much faster. For example, they know how many digits are in a valid code, and what numbers they usually start with. They often use computers to hack out the numbers for them. This kind of repetitive work is perfect for computers. Usually several good codes can be hacked out each night with a brute force, trial and error method, using a computer to dial and try the codes. Any autodial modem can be used with a simple program to check the results and re-dial. They can trade a single working number with other phreaks, so one number can be traded for several others.

Some phreaks know enough about the numbers to develop formulas to create many valid numbers from one. They still must test each number by trial and error, but most of the error is gone. Others apply these same techniques to charge card numbers, then attempt to use these fraudulent numbers to order merchandise, or charge long distance network usage.

Phone Company Numbers

Another popular phreaking activity involves investigating and manipulating the phone system. The phone company has special numbers that are used for a variety of different things. Phreakers use ANI numbers to identify the phone number of the telephone they are calling from. This may sound a bit strange, but many phreakers don't like to make illegal long distance calls from their own phones.

A loop is a circuit the telephone linemen use to test the phone lines. One phreaker calls a number that will connect him with one side of the loop, and another will call the other end. The two calls are connected, and the phreakers can talk. When they contact each other on a loop, they do not have to give out their home phone numbers. Some conference loops will allow many people to call and talk together at the same time. Phreakers use these loops for meetings.

Loops can be local or national. Phreakers use national loops to make free phone calls. One phreak will call the local end of the loop, the other will call the national end collect. The first phreaker will be waiting on the line, and readily agree to accept

special phone numbers and long distance codes. They also use 'boxes' to route long distance calls through the phone company's computers to avoid being charged for the calls (more about this under Phone Company Numbers).

2600 Magazine is a magazine devoted to computer hackers and phone phreaks. It contains all kinds of tidbits and clues for hackers and phreaks.

2600 Magazine
PO Box 752
Middle Island, NY
11953
(516) 751-2600

Phreakers have various sources for numbers used to make free calls. Some are the telephone company's own numbers, some are unsuspecting consumers access codes. Prior to the court ordered breakup of AT&T, the telephone company used to charge high rates on long distance calls (which are very cheap for the phone company), and used the money to make up for the lower rates on local services (which are more expensive for the phone company to maintain). When AT&T broke up, many companies jumped into the lucrative long distance market.

Some of these companies lease long distance phone lines from AT&T, then install switchboards and computers, and offer long distance phone service for less than what AT&T charges. A customer calls the carrier's local phone number, enters an access code, and the company places his call. The local numbers are easy to get. The hackers hack out the individual access codes, then use them when making long distance calls, so that someone else will get the bill.

Phone phreaks have developed many different techniques of hacking out these numbers. The most obvious method is the 'brute force' method, where a phreak will start with a number (like 111111) and try a series of numbers in order (111112, 111113, 111114, etc.) until he stumbles across a few which work. This isn't very efficient, but most phreaks

RED - Imitates the tones generated by a pay phone when coins are deposited. This tricks the phone company's computer into thinking that coins are being inserted. It reduces the cost of calls (even overseas calls) to 5c for 3 minutes.

PURPLE - Combines all the functions of a red and a blue into one box.

BEIGE - A device that imitates a teletype machine.

WHITE - Generates the tones equivalent to a touch tone pad. It's used for autodialing.

GRAY - Equivalent to a touch tone pad with 16 keys. It operates at 1633 Hz.

BROWN - Combines many functions of the others into one box. It always contains at least the equivalent of a purple and a gray. This is the most powerful device currently in use. It's crystal controlled, and very stable even under temperature and power changes.

YELLOW - A 2600 Hz generator, used as a simple "MUTE" device (see below).

GREEN - This is used by a person called from a pay phone to give the caller his money back. It can also make the phone collect coins, and ring back after the caller has hung up, though the call must be made from a pay phone. Some use this in conjunction with a red box, so the caller gets his quarter back.

"MUTE" - Any device used at the receiving end that makes Bell think the called party never answered, but still permitting conversation. A black box is the most famous of these "MUTE" devices.

Detecting Boxes

In the eyes of a phone phreak, these devices all have one flaw. The phone company can find the receiving party's number. Although the phone

company can't really do anything to the receiving party, they can harass them for information about the caller. Therefore it's best not to use these devices for calling your mother or your boss at work.

The phone company has recently begun implementing two new systems, known as ESS and CISS. They are making life harder for phone phreaks. ESS stands for Electronic Switching System. This system can be used to trace a phone call in a matter of seconds, without ESS, tracing takes many minutes. This forces phone phreaks to restrict the length of their calls, and that's a problem when downloading files. CCIS stands for Common Channel Interoffice Switching. This system allows the phone company to send the control signals over a separate line, instead of using tones on the voice line. In areas where CCIS is installed, a blue box will not work, unless the call is to or from an area without CCIS.

The Anatomy and Use of a Simple Black Box

Black boxes are the most heavily used because they are extremely easy and inexpensive to construct. In fact, starting with an AT&T phone, a simple black box will cost under than $5.00. Of course, using a black box is a criminal offense, so it's not advised, even for educational purposes.

All that's required to build a black box is an SPST toggle switch, two 6" strips of wire, and a 1/2 watt, 10% resistor. The phreak gets his parts from any electronics store, and solders the two strips of wire to the switch. He then removes the bottom of the phone and the plastic case to find the network box. The network box is in the approixmate center of the box, and has labled terminals with wires attached. It's a simple matter for him to attach the resistor between the "F" and "RR" terminals, and connect one of the wires from his switch to the "RR" terminal. Now all that remains is for him to disconnect the wire that originally ran to the "F" terminal, and attach it to the other side of his switch, which is then run out of the back of the

phone.

Once the assembly is complete, the phreak will set the switch to the NORMAL position (where a dial tone can be heard) and arrange for a friend to call him long distance at a specific time. When the phone rings, he lifts and drops the receiver as fast as possible (to stop the phone from ringing), then flips his switch, and picks up the phone to talk free of charge. When he's done, he hangs up the phone, and flips the switch back to the NORMAL position, ready for it's next use.

WARNING - The phone company can randomly check for black boxes. Persons caught using these devices may be subject to criminal prosecution.

How It Works

When someone calls long distance, the billing starts when the phone is answered. The phone company knows the phone was answered, because a return voltage begins to flow as soon as the receiver is lifted. The resistor in the black box cuts this voltage down to a point where it is too low for billing to begin, but still high enough to work the mouthpiece. When the receiver is quickly lifted and dropped, the ringing is stopped, but the receiver is not off the hook long enough for billing to start. If the receiver is lifted for one full second, billing begins, and the call will be disconnected when the phone is hung up and the switch is flipped.

More elaborate black boxes are fully automated, so timing is not important, but they are much more complex to build.

Cracking

Cracking is illegally accessing on-line computers. The movie 'War Games', about a teenager who breaks into the Defense Department's computer system, brought cracking into the limelight. cThere was a large increase in cracking activities after the release of the movie, but it also served to alert systems operators to the dangers. Cracking is closely related to on-line protection, which is discussed in detail in Chapter 7.

Some amateurs were surprisingly successful. Six

months after the movie was released, a dozen people in Milwaukee, WI, ranging in age from 15 to 22, broke into computers in a nuclear weapons laboratory in Los Alamos, a Los Angeles bank, and a dozen other firms in the U.S. and Canada. One of the group members said "It didn't take too much intelligence to get into the things".

Like other computer users, crackers and hackers form clubs. Bill Landreth, a.k.a. The Cracker, and a friend formed a now famous group, The Inner Circle. He began breaking into systems when he was fourteen, and was caught by the F.B.I. when he was seventeen. In his book, Out of the Inner Circle, he tells of his first experience in cracking into a system. A friend had given him the number of a local firm's computer. He called the system, and began trying passwords. He called twice and tried using first names, with no luck. He says "My third try was LEE. Against odds no gambler would ever bet on, it worked... three tries with no clues, and I hit on a valid account/password combination."

Most hackers are not destructive, they just enjoy exploring large computer systems. If they have the time, they will often spend as much as 60 or 70 hours a week hacking. They may crack only four or five new systems each year, and most of these accounts will die within six months. By trading information, a hacker can gain access to two dozen or more different systems. Most hackers despise people who destroy data or files in computer systems. When a user ID and password are abused, the system's personnel will discover the damage and cut off that account. The hacker will no longer be able to use the system unless he can crack it again.

Hackers will often set up private bulletin boards, to post and exchange information and messages. An experienced cracker is a security expert, so these boards usually have elaborate security systems (see on-line protection). This kind of bulletin board was the primary communication channel between the members of the Inner Circle. Less exclusive hackers will trade information over public bulletin boards, and occasionally over public information services, such as Comp-U-Serve.

In his book, Bill describes the different kinds of hackers. 'Novices' are attracted to hacking because it seems like fun mischief. They usually don't get too far, and quickly become bored. 'Students' enjoy exploring the system, and learning as much as they can about the way it operates. 'Tourists' enjoy the challenge of breaking in, once they have succeeded they are usually not interested in exploring the system any further. 'Crashers' are deliberately destructive, their sole intent is to see how much damage they can cause. Most other hackers don't like 'crashers'. A 'thief' will gain access to a system to steal valuable data. Often he works for the company he is stealing from.

For people who want to be hackers, but don't want to break the law, Activision has a partial solution. In September 1985, they released a game called Hacker, which simulates a computer break-in. It comes with virtually no instructions, and when booted, presents the player with the message "Logon Please". The object of the game is to gain access to the system and discover illegal actions by the company which owns the system. The game scenario is actually nothing like a real system, but it may be a refreshing change of pace.

PIRATE BOARDS

The Front Operation

Bulletin boards are a popular way for many pirates to exchange software. Since piracy is illegal, many pirates hesitate to post notices referring to piracy or illegally copied software on public information services, such as Comp-U-Serve. A pirate who leaves a message on an independent bulletin board risks the wrath of the sysop (system operator). Because of the dangers in 'open waters', pirates set up their own bulletin boards.

At first, a pirate boards looks just like any other board, except that it seems very small. There's a few old public domain files to download, and a few old messages. It appears to be a very dull system. Most browsers log on, look around and leave. They've seen the 'front operation', but they cant't

get at what lies behind it . The front is a cover, set up to discourage curious tourists from examining the board any closer.

Pirate boards usually have layers of security, so without an access code, all a caller will see is the front. An inquisitive caller may wonder how a board so empty can have four full disk drives, but unless he tries to crack the system, or leaves an intriguing message for the sysop, he'll never know what's there. With the right access code, a user can get past the front. The higher the users' security level, the more he can access on the board. Some boards have many different levels, so only pirates with the highest codes have access to the most valued software.

Inside a Pirate Board

A pirate board is the software pirates treasure chest, and the board's phone number is the key. The boards post listings of other pirate boards, chaining together a worldwide 'underground' network. Some postings include specially worded messages; 'keys' for new pirates to get past the front on other boards.

The biggest prize among pirates is software, and large, popular boards, or boards visited by insiders, have plenty of it. Most of the files are broken copies of programs, because protected programs are hard to transmit (see transmitting protected programs). This means a pirate can make an unlimited number of copies of programs he downloads.

In addition to programs for downloading, most pirate boards also post listings from the sysop and other pirates willing to trade. These listings are called Want Lists and Available Lists. Want Lists are lists of software that a pirate, or the sysop is looking for. Software on a Want List is usually something a pirate will go to extremes to try to get. They are almost always brand new, unreleased titles, or rare older programs. The Available List shows what the pirate has to trade for items on his Want List. He may often trade many older programs for one hot new title.

Pirates also use a section of their boards for

Classified Ads to buy, sell, or trade. Some pirates buy blank disks from mail order houses, in large enough quantities to get discounts. They will then offer the extra disks for sale, at prices far less than most computer stores can afford to charge. Some pirates sell used software, others post ads to buy or sell used hardware. A few will sell pirated copies of software, or photocopies of documentation.

Pirate boards are often used by hackers, crackers, and phone phreaks to communicate and exchange information. Crackers will post phone numbers of computer systems. Occasionally, they will post listings of IDs and passwords for Comp-U-Serve, other public information services, public or private bulletin boards, and other private computer systems. Most publicly posted IDs are abused, so when these are left, it's usually on one of the highest security levels. Crackers will also leave clues for other crackers who post messages asking for help.

These boards often have lists of various numbers. Sometimes there are lists of AT&T calling card numbers, sometimes they have lists of access codes for MCI or other long distance carriers. A few even post lists of stolen credit card numbers.

Phone Phreaks often trade information over pirate boards, posting the newest information on black boxes and other hardware. Occasionally, information on cable descrambler boxes will appear. A user who needs help building a box can get plenty by leaving a message.

Recently even the FBI has gone into the BBS business. They have set up systems to catch law breaking users. These 'sting' operations are discussed in 'Cracking Down on Sysops'.

What's in a Name? Naming to Deceive and Trojan Horse Programs

You can't judge a book by it's cover, and you can't judge a program by it's name. A popular practical joke is to take a less than popular program and palm it off as something it's not. It's disguised as a hot new program that many traders would like to get. For example, a pirate would choose a popular new video game, and find an unprotected copy of another

game based on the same idea. He would then rename the file so it has the same name as the new game. An elaborate pirate might go so far as to change the introduction screen to display the new name. He would then upload the program in exchange for other software, or post it on an available list, and try to trade it with other pirates.

A far more dangerous kind of deception occurs with programs downloaded from bulletin boards. These 'Trojan Horse' programs are described as performing one task, but really do something quite different, and usually something destructive. One such program has appeared on several bulletin boards around the country. A New Jersey executive had a disastrous experience with the program, written for IBM computers. At the time, he had over 900 programs stored on his 20 meg hard disk. He logged onto a local bulletin board one night, and downloaded a program which promised to improve his computer's graphics. He sat back to watch the program work. Instead of great graphics, all 900 programs vanished, and all that was left was a simple message - "Arf! Arf! Got You!".

Bill Machrone, editor of PC Magazine, says these Trojan Horse programs have been popular college campus jokes for quite some time. Other 'jokes' include programs with 'worms' (discussed in Chapter 9), or programs that will work two or three times, then gobble programs or format disks the fourth time they are used.

Cracking Down on Sysops

Electronic communication is a brand new area with a vast and untapped potential. Because this area is so new it currently resides in a legal limbo, as the courts and legislatures struggle with the legal problems posed by this new form of communication.

Freedom of speech and of the press is guaranteed to all American citizens by the United States Constitution. But even this broad protection comes with some limitations. It is illegal to publish newsletters of stolen charge card numbers, and it's illegal to mail such a newsletter to subscribers. Yet something very similar happens hundreds of times every day all across the United States. Instead of

being printed on paper and sent through the U.S. Mail, it happens electronically through bulletin boards. This poses an obvious dilemma for law enforcement officials, sysops, and innocent victims.

Newspaper editors are held responsible for what their newspaper prints, so a logical solution would be to hold sysops responsible for the material that appears on their boards, but the matter is not so simple. First, a sysop cannot censor all electronic conversation on his board at all times. It is unreasonable to expect a sysop to check each new public message each day for potentially offensive or illegal information. Many boards allow multiple users to be logged on simultaneously, and no one person could possibly monitor all that. Even when a sysop makes every reasonable attempt to censor public messages, the question of what to do about private messages remains. A private message should be read only by the person to whom it is addressed. Anything else is a violation of the sender's rights. Postal employees are not allowed to read conventional mail, electronic mail users should have the same rights.

Censoring does not solve the problem. Censoring does create an added burden for honest sysops, and threatens to thwart the growth of electronic communication.

The sysop is in a no-win position. On one hand, he must respect his users' right to privacy. On the other, sysops have been held liable when illegal information has been posted on their boards. In one case involving the posting of a stolen charge card number, the sysop was convicted on criminal charges, and later was sued for civil damages by the person whose number had been posted.

To protect the telecomputing population and innocent sysops, several bills protecting electronic information from unwarranted searches and seizures have been proposed (see New Trends in Software Law). Most are designed to provide telecommunications with the same protections afforded voice phone calls under the Federal Wiretap laws, and permit law enforcement officials access to private telecommunications only when probable cause

is present.

Pushed into a corner by tougher legislation protecting electronic communication, and pressure to crack down on computer crime, law enforcement officials have begun to use tactics which some people find questionable. The Austin, Texas Police Department set up a bulletin board called the Underground Tunnel. Sgt Robert Ansley, calling himself Pluto, ran the board for over two years. Using the information obtained from messages left on the board the police department closed two pirate boards, though no one has been arrested because of involvement with the sting operation.

The police department insisted they had been extremely careful to avoid solicitations or entrapment of any kind, and stated that the messages on the board had been scripted in conjunction with the district attorney's office. Still, many innocent users were startled to learn the board had been an undercover police operation, and were understandably concerned about the privacy of mail left on the board.

The revelation that the Austin board was a sting was startling, but police investigations of bulletin boards is rapidly becoming widespread. Police in Fredmont, California, arrested eight persons for credit card fraud, misuse of telephone credit card operations, and technical tresspass after a 3 1/2 month bulletin board investigation.

Congress is also considering stricter laws for sysops. The Computer Pornography and Child Exploitation Act (S. 1305) would make it illegal to transmit obscene material specifically pertaining to the sexual exploitation of children, and set up penalties for sysops who knowingly engage in that activity. Naturally, this bill would not affect the vast majority of bulletin boards. Legislators across the country are considering many other computer laws, including many which affect telecommunications. These laws and their ramifications are discussed in full in the chapter 'New Trends in Software Law'.

PROBLEMS FOR PIRATES

Documentation

Lack of documentation can be a deterrent to pirating in some situations. Game software usually doesn't require any documentation to be used successfully, but utilities and productivity software often do. Printed documentation can be easily duplicated with a photocopy machine. An office employee with access to a copy machine can make an unlimited number of free copies. Inexpensive personal copy machines are becoming more and more popular.

Printed material however, cannot easily be copied and sent over the telephone lines, it must be mailed. From a pirate's perspective, this isn't a very convenient, or reliable, method of exchange.

Documentation in the form of a word processor data file is easy to transmit over a modem along with a program. The problem with this method is that the entire documentation must be typed in, and pictures and diagrams are ususally lost. Some clever pirates have used graphics characters and picture files to avoid this pitfall. In spite of the problems involved in converting printed documentation into a disk file, they are fast becoming available on many pirate boards. Some documentation files are as highly prized as the program itself.

Transmitting Protected Programs

Using modems to transmit files to and from Atari computers is a popular pastime. Sources like Comp-U-Serve and Atari bulletin boards transmit thousands of files every day to locations throughout the country. Many software pirates use modems to trade programs over long distances. These programs are usually broken down into unprotected files and transmitted using X-Modem or Teletalk protocols. A whole disk of data can also be transmitted on a sector by sector basis using a public domain program such as DISKFER, or the program can be packed down into one large file, transmitted, and expanded at the other end. A serious drawback - from a

pirates point of view - of transmitting programs this way is that it's not easy to send a protected program. Even a program with simple bad sectors cannot be transmitted with any common software.

One solution pirates use is to send the whole disk in the mail. This is certainly the easiest solution, but many pirates dislike waiting for the disk to arrive, and worry about accidental damage while the disk is in the mail. Pirates also expect programs in exchange, and distrust other pirates.

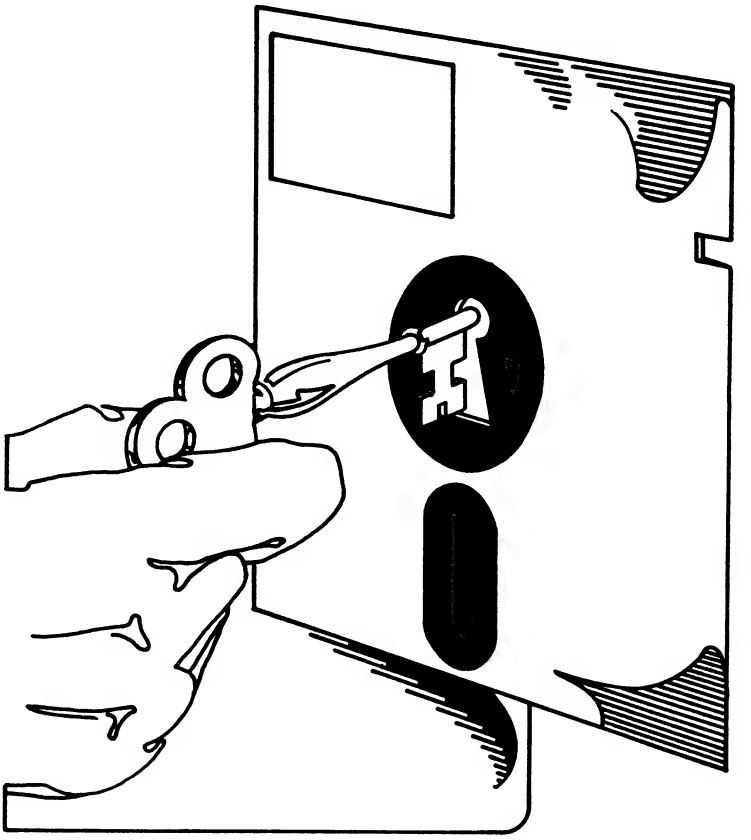
Transmitting a program via modem overcomes these problems, but this can be expensive if it's long distance. It is possible to cut down on the transmission time by using a data compressor to shrink down the program. Then it can be transmitted and re-expanded at the other end for use.

Of course, some people use phreaking methods (see Phone Phreaking section) to avoid these charges altogether. For them, transmitting a program is an excellent solution.

There are only a few ways to transmit a protected program. Although commercial products to do this have been offered, none have been delivered at this time.

The simplest way to send some protected programs is with the aid of a hardware device such as the Archiver/Editor (a version of this software runs on Happy also). At the transmitting end, the user changes all bad sectors (and sectors with bad status) into good sectors, then transmits the entire disk. The receiver must then reapply the protection as instructed by the sender. This works for some programs, but gets more complicated when duplicate, short, or unstable sectors are involved. To handle these protection methods, the sender must capture any additional information from the disk (like that contained in duplicate sectors), and send it correctly. The receiver must then create the proper custom format on the disk, and move the data in to make it work. Considering the difficulties of doing all this manually, it's surprising that no one has yet produced a good program to do this automatically.

SECTION II
THE MEDIA AND THE METHODS



INTRODUCTION

This section discusses the methods of software protection. The protection methods are divided by the media on which the software and protection code is stored; disks, cartridges, hardware data keys, on-line protection, and general and miscellaneous methods. Each segment discusses the techniques used by both pirates (to break the protection) and software publishers (to prevent illegal copies).

All forms of protection are intended to do one or both of two things. First, and perhaps most importantly, software protection methods are created to prevent users from making and/or distributing illegal copies. The second reason is to prevent the user from listing and modifying the program. The goal here is to protect programming secrets. Software code protected in this fashion can be classified as a Trade Secret. This also prevents others from copying routines (or the entire program), making minor modifications, and then marketing the slightly modified version. The methods used to achieve both these goals are discussed in this section.

Chapter 5

DISK SPECIFIC PROTECTION

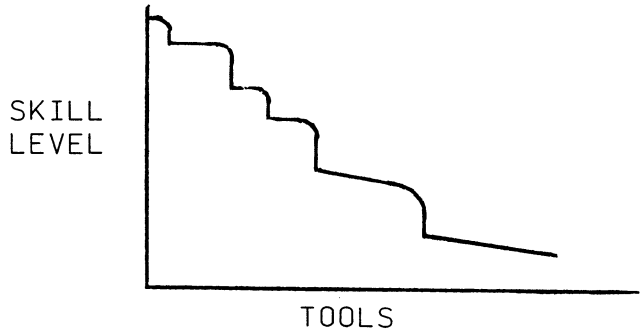
Over 80% of the commercial software sold for Atari computers is on disks. Disks have become the media of choice because of their low cost (compared to cartridges), high speed and reliability (compared to tapes). This book will discuss disk protection first because disks are so important in the software industry. It will detail general disk protection and copy methods. Disks also will be covered further in "The Tools" which analyzes and compares disk and other copy tools. Since disks are also the media of choice among pirates, the special software which comes with this package contains several utilities for protecting disk programs. Overall, disks offer many unique opportunities for sophisticated protection techniques and semi-automated disk back up systems.

HOW PIRATES COPY DISKS

Skills vs Tools

The methods used to back-up protected disks vary with the skill of the user, the back up tools available, and the complexity and sophistication of the protection in the program. In the time since Volume I of this series was written, software publishers have enhanced their protection techniques to the point where virtually no foolproof software back up methods exist. Backing up software protected with the newest techniques requires either a user with a good understanding of software protection and plenty of tools, or a very advanced, very talented programmer, and one or two good

tools. Unless a revolutionary new copy device appears on the scene, the days when a novice with a few tools could make back up copies of any commercial program are over.



As the accompanying diagram shows, the right back-up tools (such as those described in "The Tools") can greatly reduce the skill needed to back up software. Still, the need for skill is never completely eliminated, particularly if the program involves intricate, inventive protection schemes. The amount of skill involved in creating a back up copy has a tremendous effect on the final result. An advanced hardware based back up tool can enable a user to create a working back up copy with a minimum of skill and effort, but the back up copy will itself be copy protected with original copy protection scheme, and just as difficult to back up. On the other side of the scale, a talented programmer can, with time and effort, remove the protection completely. The result is a version which is both easy to duplicate, even for a novice, and easy to transmit over a modem. Naturally, most pirates prefer an unprotected copy. Many other users also feel an unprotected version of a program is superior to a protected version. They may wish to store it on a specific disk with other programs, or create several back-up copies to keep 'just in case'.

Skilled users try to modify programs to suit their needs. Another advantage of an unprotected copy is that it loads faster, and causes less wear and tear on a disk drive.

Software protection and back up techniques are so closely related, it is difficult to understand one

without the other. To overcome this difficulty, this book will discuss both techniques together when possible.

Steps in Creating Backups

Every back-up method used by pirates contain one or more of the following steps.

1. Load Analysis - This involves observing the program as it loads, watching and listening for the protection. It helps to use a disk drive which indicates the track numbers (eg. RANA or Indus GT). Normally the drive will slow, or even stop completely, when it's checking the protection, but this is not always the case.

2. Sector Analysis - In this process, the pirate determines the statuses received from the sectors on the disk, and searches for duplicate sectors and other special formats. This can give a pirate a clue as to what to look for in the code, and possible back-up methods.

3. Directory Analysis - The first step is to decide if a directory is present. If a directory is used, the pirate will see if it is hidden or normal. A directory scanner, such as the one in Scanalyzer, can help here.

4. Program Analysis - This process involves several steps. The program itself determines what steps are used.

A). If the program is in BASIC

- Test if it is listable.
- If not, test if a Detokenizer or Variable Fixer will work.

A detokenizer analyses each byte of an unlistable BASIC file and lists the commands, variables, and numbers they represent. Some can also insert corrected variable names if needed. Disk Doctor and Scanalyzer are two examples.

B). If the program is in machine code

- analyze the boot sectors and the loader.
- Use a disassembler to study the program.
  A disassembler reads the data in a file and converts it to the Assembler Language equivalent. This makes the data much more readable, and infinitely easier to follow. Super Disassembler, Diskey, and Scanalyzer are examples.
- Watch for direct disk access done by the program.

C). If possible, interrupt the execution of the program and study the code as it resides in RAM.

5. Protection Duplication - In this process, the pirate tries to re-create the protection used on the original disk on a back-up copy. Special hardware is required to duplicate most sophisticated protection schemes. Popular hardware includes the Happy drive modification, the Archiver, the Impossible, etc (see The Tools). Sometimes an alternate scheme can trick the protection code and make the program run. This is useful when the original protection cannot be re-created.

6. Alter Protection Code - This is modifying the programs so they no longer need the protection to run. This process can be extremely complicated, but if successful, yields a completely unprotected copy.

DISK PROTECTION BREAKDOWN

Many various techniques for protecting disks were discussed quite thoroughly in Vol I. Rather than repeat that discussion here, this section offers a brief outline of those techniques covered in the first book. The newest techniques, developed since then, are covered in detail in the next section of this chapter. Read this section to review these techniques before continuing to the latest, more difficult techniques in the next section. If you are

unfamiliar with any of the older techniques below, you should refer to Vol I for specific information before reading the rest of this chapter.

Creating Non-LISTable/Modifiable Code

Preventing listings and modifications of program code can go a long way toward stopping piracy. Some of the techniques used to accomplish this are:

1. Disabling the Break Key and the System Reset Key with POKEs. This is discussed in Vol I.

2. Preventing Error Breaks and LOAD and SAVE combinations in BASIC programs. - This includes TRAPing errors and making the program unLISTable (with POKEs), and/or hiding the disk directory. Hiding the disk directory or changing the VTOC (Volume Table of Contents) can prevent a standard DOS copy, but not a sector copy. Specific information on advanced directory hiding can be found in the next section of this chapter.

3. Hidden Variables. - This was a popular method of protecting BASIC programs prior to making programs unLISTable. The table of variable names is wiped out by changing them all to CHR$(155), the return character.

4. Compiling BASIC programs - This process involves using a BASIC compiler like ABC from Monarch Data Systems, or DataSoft's BASIC Compiler. It makes the program unLISTable, and also run much faster. The drawbacks are that the program may take up more space and memory, and can be very difficult to debug.

5. Making Assembler unLISTable by inserting extraneous BYTE commands. - This method involves the use of BYTE commands spread throughout an Assembler program. This makes the program much harder to disassemble and list, because it makes the data look like instructions.

6. Self Modifying Code - In this method a program uses innocent looking commands to change other commands into new code, such as disk reads, after the program has begun executing. These changing instructions are very difficult for a pirate to crack.

7. Encryption - This method of protecting a program involves encoding the finished program to confuse a disassembler. The disk included with this package contains a powerful encrypter program that not only stops a disassembler, but also prevents people from reading or changing literals in a program (such as the authors name, copyright, etc.). See the section in this book on data encryption for more information.

Preventing Copies - These methods protect a disk by creating a custom format which the program checks for when it's running. If the format is not found, the program can partially run, or stop altogether.

Bad Sectors - are sectors which contain unreadable data. Creating standard bad sectors does not require special hardware.

CRC Errors - are soft errors which occur when the CRC bytes do not match the data. These sectors can be checked for a bad status and/or special data.

Bad Data Marks - are data marks other than the standard $FB and also cause a soft error.

Duplicate sectors - are two sectors which have the same sector number but contain different data. The program checks for this by repeatedly reading the sector, and comparing the results.

ADVANCED DIRECTORY HIDING

The basics of disk directories and directory

hiding are explained in Vol I. This section covers advanced directory capabilities and techniques including:

1) Full range directory hiding.

2) Hiding directories with DOS 2.5.

3) Pseudo directories and partial directories.

A program is included on the Advanced Protection Techniques disk which will do some of these things.

Whole Disk Range and DOS 2.5

Volume I explains how to hide a directory anywhere from sector 256 to sector 510 by modifying location 4226 in memory. The restrictions on the range of sectors exists because 255 is the biggest value that can be poked into a single location. Since this method relies on a modified DOS to find the hidden directory, it works only on DOS 2.0.

To hide a directory anywhere on the entire range of sectors requires modifying 2 bytes. The two bytes are handled the way Atari handles all numbers, in a low-byte, high-byte order. These two bytes taken together make up the sector number where the directory begins. To compute the bytes, use this formula:

DIRECTORY STARTING SECTOR = (HIGH-BYTE * 256) + LOW-BYTE

The high- and low-bytes are stored in separate places depending on the version of DOS you are using. In DOS 2.0 the low-byte is stored in location 4226, and the high-byte goes in location 4229.

As an example, suppose you want to use a directory that starts in sector 710. To do this, poke 4229 (the high-byte) with 2 and poke 4226 (the low-byte) with 198, because (2*256)+198 = 710. The procedure to access a hidden directory under DOS 2.5 is the same except the high-byte is stored in location 4274, and the low-byte goes in location 4171.

Using the techniques described in Vol I and adding the procedures described above, you can hide or access disk directories anywhere on a disk.

Pseudo and Partial Directories

Vol I explains the steps involved in preparing a hidden directory. The last step says to wipe out the original directory stored in sectors 361-368. If that method is used, a regular DOS will show an empty directory.

Sometimes you may wish to make certain files accessible, but leave others hidden, or perhaps put a dummy directory in the normal location to throw off any pirates. This is relatively easy to do.

Using a sector editor such as Diskey, Scanalyzer, etc., edit sectors 361 through 368 and modify them as you wish. Each directory entry contains the following information:

Byte	Function
0	Indicator flag $00 represents an empty entry that was never used. $01 means the file is open for I/O. $02 means the file was created with DOS 2. $20 indicated a locked file. $40 indicates the file is in use. This is the normal entry. $80 indicates the file was deleted.
1-2	2 byte number indicating the number of sectors in the file.
3-4	2 byte number indicating the starting sector of the file.
5-12	the file name.
13-15	the file extender (the letters after the period).

The simplest modification is to change the starting sector. This will cause a normal DOS to get an error 164 if the file is read.

Another approach is to copy the entire directory from a different disk. It could show some files and not others, or be totally empty, etc. Be sure to

correct any entries for files that you want the user to be able to access, and always test it to be sure the changes are what you wanted.

OVERFILLED TRACKS

One of the newest and hardest to duplicate disk protection techniques is overfilled tracks. First appearing in early 1985, this protection method defied all the normal copy methods.

A standard single density Atari track holds 18 sectors, of 128 bytes each. However, a standard drive can actually read more information than that. The limit is about 21 sectors of 128 bytes, or even more sectors if some of them contain less than 128 bytes.

One thing to note is that a standard drive can only address a total of 18 sectors on a track, so any protection method using more than that must also use duplicate sectoring (explained earlier). If the sectors are set up with less than 128 bytes, then more than 21 can fit onto a track (short sectors are explained later in this chapter). No protection method to date has used more than 26 sectors, but the limit is actually much higher.

In order for a program to check for more than 18 sectors, it must utilize the same techniques as those used to detect duplicate sectors. That is, read a sector more than once, so that all the sectors with a matching sector number can be found. Therefore, programming to check for overfilled tracks is really nothing new, what's different is the amount of data contained and the way the track is created.

The important thing about overfilled tracks is that commonly used copy devices couldn't automatically copy or create them. In other words, this method could stop all but the most creative pirates.

Creating Overfilled Tracks

Since an overfilled track requires a special custom format, only a modified drive can create one. But even a custom drive couldn't write that

much data to a single track. The drive head just couldn't write fast enough to fit it in. In addition, none of the copy software had big enough buffers to hold all this information, so attempts to write it back were futile.

Software companies who employed this method (Synapse, Electronic Arts, etc.) created the tracks on very specialized and expensive high capacity drives which pirates don't have access to. Once again, it looked like the software companies had achieved a protection method that couldn't be copied by off-the-shelf hardware and software, and their only hope was to break it by hand.

Copying More Than 18 Sectors Per Track

Ironically, the solution the pirates came up with very closely resembled the first method discovered to create bad sectors. Once again, it was found that slowing down the speed of the drive was the assistance needed to get the job done. The first use involved slowing the drive down to 220 RPM to create bad sectors as detailed in Vol I. To create an overfilled track the drive can be slowed to 269 RPM (from the normal 288). This is just slow enough to write some extra data, while fast enough to avoid bad sectors. Of course the hardware still has to be able to create duplicate sectors, but several off-the-shelf modifications can do that. This method works with up to about 21 full sectors per track, but by using tricks discussed later, software companies have gone beyond that, so new methods were needed.

This technique of slowing down the drive became popular enough that Happy Computers supplied information explaining how to do it in one of their software releases. They call it the H.W.A. mode (Happy Wins Again) and it simply automates the process of slowing the drive down. (This tricky modification has been dropped in favor of their new Pre-Defined Back-up files, which will make backups of specific programs. See the review of Happy Enhancement for more information.)

SHORT SECTORS

The cases where more than 21 sectors per track are used are few, but usually they contain many short sectors. A short sector is just like a normal one except that it contains less than 128 bytes. Once again special hardware is needed to create short sectors, but copying them is more of a trick than ever, because of the way a disk drive reads a sector.

The Workings of a Short Sector

When a disk drive reads a short sector, it still returns 128 bytes to the buffer even though it may include the next sectors in the track. In other words, the drive finds the start of the sector by reading the header information, then continues to pass back the next 128 bytes no matter what they contain.

What this means is that a copy program will read more data from a disk than was actually written. For example, say sectors 1, 2, and 3 are in order on a disk and contain only 10 bytes of data each. (A sample diagram is shown in the section on re-creating protection). When the drive reads sector 1 it will pass back the 10 bytes of data from sector 1, then continue to pass back sector 2's header block (44 bytes which tells the drive sector 2 starts here), sector 2's 10 bytes of data, sectors 3's header block, the data from sector 3, and then part of sector 4's header block, a total of 128 bytes. When sector 2 is read, the drive will pass back it's data plus the header and data bytes from sectors 3 and 4. This poses a serious problem when you attempt to copy the disk. The copy program doesn't know that the sectors weren't full, so it attempts to write out all the data it read. This can be significantly more data than can be successfully written to the disk with the hardware.

Short Sectoring has been used to put up to 26 sectors on a track even though it only contained about the normal amount of data. Attempting to copy this by putting back 26 full sectors can't work, for the reason explained earlier.

Copying Short Sectors

No currently available hardware (including Happy's HWA mode) can automaticaly copy short sectors (more than 21 sectors per track), but some are capable of re-creating the track if the user knows how, and one can trick some programs into running without them. The techniques used to re-create short sectors are discussed in the section Re-Creating Protection Techniques in this chapter.

UNSTABLE SECTORING (PHANTOM AND FUZZY SECTORS)

Perhaps the newest method employed by publishers is unstable sectoring. An unstable sector, sometimes referred to as a phantom or fuzzy sector, is a disk sector that seems to change each time it is read. Even though the sector physically occurs only once on the disk, it behaves like multiple duplicate sectors. Each successive read shows the data contained in the sector to be different.

Using unstable sectoring to protect a program is simple. Just as with duplicate sectors, the program can read the sector twice and compare the results. If the results are different, then the program would run as usual, but if the results are the same, the program knows that it is a copied disk and can lock-up or self destruct.

The real power in unstable sectoring lies in it's ability to fool disk modifications. When a drive reads an unstable sector it has no way of knowing the results will change with each successive read. It then writes out the data it read onto the users back up disk, but of course the sector that it writes is stable, and will not change no matter how many times it is read. In other words the drive and the copy program accurately copy the data and status of the original sector to the back-up disk, but when it does so it creates a stable sector that can easily be detected by the program.

The Workings Of An Unstable Sector

An unstable sector gets it's characteristics from it's magnetic makeup. Instead of an orderly array of magnetized areas on the disk representing zeroes and ones, an unstable sector contains an area that is unformatted.

The unstable sector has a normal header block identifying the sector to the disk drive, then at least a few formatted bytes of data. After that, however, the magnetic landscape is a mess. It is this unformatted area of the sector where the drive reads the seemingly random data.

Copying Unstable Sectors

Fortunately for software producers, there is no simple or direct way to copy an unstable sector 100% of the time. Some special programs can know the specific protection on a program, and create a patch track (like Happys Pre-Defined Backups) and others (like The Impossible) can, under the right conditions, trick the program into running anyway. However, these methods work only in limited cases. As with short sectoring, the best way to make back ups is through breaking them by hand or attempting to re-create the format or an acceptable alternative. Either way it takes patience and skill to successfully back up these methods. The best alternative is to use the methods described in the next section Re-Creating Protection Techniques.

RECREATING PROTECTION TECHNIQUES

Software publishers have added some new tricks to their arsenal of protection methods. These methods (short sectoring and unstable sectoring) fool the disk drive and the copy program into thinking that the disk is formatted one way, when it is really formatted in a very different way. When a user attempts to back up these programs, he is usually doomed to failure. For the time being, these protection methods can't be re-created automatically. The formats must be studied, and

then rebuilt from scratch.

The most useful tool we've found is the Archiver/Editor Software (see review in The Tools section). First, the user determines the track layout, then uses the formatter option of the Archiver/Editor to recreate the same layout on the back up copy. Using this method, and a little creativity, it is possible to recreate all the disk protection methods currently in use.

Mapping the Track

The first step to mapping the track is to read the track several times, then study and compare the results. The most obvious thing to look for is a change from one read to the next. On each read, write down the sector numbers found, the statuses, and the number of duplicates (sectors with the same number). Also, scan the data noting any fill bytes. Fill bytes are a single character that fills all or part of a sector.

When working with overfilled tracks or short sectors, it may take several reads before all of the information is actually read. The whole track often overflows the buffers of the read program, so that only part of the track appears on each read. You must be sure that all sectors and their duplicates have been found in order to successfully recreate the protection. Also, watch the sector data closely for short sectors. Short sectors start out normally, but then go on to contain the header and data bytes for subsequent sectors. The trick is to look at the data in a sector and check if that data is found in a previous sector.

An example of short sectors is provided in diagrams A, B, and C. In this example, sectors 1, 2, and 3 are in order on the track, and contain the following data:

SECTOR	LENGTH	FILL DATA ($)
1	10	AA
2	10	BB
3	128	CC

SECTOR 1

BYTE #	HEX DATA	ASCII DATA
0000:	AA AA AA AA AA AA AA AA	********
0008:	AA AA 0E EB FF FF FF FF	**.k
0010:	FF FF FF FF FF 00 00 00	...
0018:	FF FF FF FF FF FF 01 D8	.X
0020:	FB FD FF C2 CC FF FF FF	() BL
0028:	FF FF FF FF FF FF FF FF	
0030:	FF FF FF FF FF FF 04 BB	.;
0038:	BB BB BB BB BB BB BB BB	;;;;;;;;
0040:	BB 62 A5 FF FF FF FF FF	;b%
0048:	FF FF FF FF 00 00 00 FF	...
0050:	FF FF FF FF FF 01 D8 F7	.Xw
0058:	FC FF 84 9C FF FF FF FF	| ..
0060:	FF FF FF FF FF FF FF FF	
0068:	FF FF FF FF FF 04 CC CC	.LL
0070:	CC CC CC CC CC CC CC CC	LLLLLLLL
0078:	CC CC CC CC CC CC CC CC	LLLLLLLL

SECTOR 2

BYTE #	HEX DATA	ASCII DATA
0000:	BB BB BB BB BB BB BB BB	;;;;;;;;
0008:	BB BB 62 A5 FF FF FF FF	;;b%
0010:	FF FF FF FF FF 00 00 00	...
0018:	FF FF FF FF FF FF 01 D8	.X
0020:	F7 FC FF 84 9C FF FF FF	W: ..
0028:	FF FF FF FF FF FF FF FF	
0030:	FF FF FF FF FF FF 04 CC	.L
0038:	CC CC CC CC CC CC CC CC	LLLLLLLL
0040:	CC CC CC CC CC CC CC CC	LLLLLLLL
0048:	CC CC CC CC CC CC CC CC	LLLLLLLL
0050:	CC CC CC CC CC CC CC CC	LLLLLLLL
0058:	CC CC CC CC CC CC CC CC	LLLLLLLL
0060:	CC CC CC CC CC CC CC CC	LLLLLLLL
0068:	CC CC CC CC CC CC CC CC	LLLLLLLL
0070:	CC CC CC CC CC CC CC CC	LLLLLLLL
0078:	CC CC CC CC CC CC CC CC	LLLLLLLL

SECTOR 3

BYTE #	HEX DATA	ASCII DATA
0000:	CC CC CC CC CC CC CC CC	LLLLLLLL
0008:	CC CC CC CC CC CC CC CC	LLLLLLLL
0010:	CC CC CC CC CC CC CC CC	LLLLLLLL
0018:	CC CC CC CC CC CC CC CC	LLLLLLLL
0020:	CC CC CC CC CC CC CC CC	LLLLLLLL
0028:	CC CC CC CC CC CC CC CC	LLLLLLLL
0030:	CC CC CC CC CC CC CC CC	LLLLLLLL
0038:	CC CC CC CC CC CC CC CC	LLLLLLLL
0040:	CC CC CC CC CC CC CC CC	LLLLLLLL
0048:	CC CC CC CC CC CC CC CC	LLLLLLLL
0050:	CC CC CC CC CC CC CC CC	LLLLLLLL
0058:	CC CC CC CC CC CC CC CC	LLLLLLLL
0060:	CC CC CC CC CC CC CC CC	LLLLLLLL
0068:	CC CC CC CC CC CC CC CC	LLLLLLLL
0070:	CC CC CC CC CC CC CC CC	LLLLLLLL
0078:	CC CC CC CC CC CC CC CC	LLLLLLLL

As you can see, sector 3 looks normal and is filled with $CC. However, sector 2 is only 10 bytes long, so when it is read you get 10 sets of $BBs, then you see the sector ID data that usually falls between sectors, and finally you see the $CCs that make up sector 3. In other words, the drive reads the 10 bytes of sector 2, but then goes on to get the following data (even though it is not part of the sector), up to 128 bytes.

Sector 1 shows its 10 bytes of data containing $AA, then goes on to find sector 2 and part of sector 3. When you see this kind of data, it should be apparent that you are looking at short sectors.

Once you have identified which sectors are short you can recreate them using the Formatter screen in the Archiver software. Just fill in the length bytes and the fill data and it lets you easily write up to 24 sectors per track. As explained earlier, these sectors could not be written normally because there is too much data to fit.

Conclusion

This basic method of analyzing then creating the custom track can be used to recreate virtually every protection technique used today. Although it requires some thought and skill, it is still easier than patching the code. It does yield a copy that is as protected as the original. With an understanding of these methods, and a little creativity, a programmer can develop his own custom protection methods. Unfortunately, it also means that a pirate can copy anything available.

Chapter 6

ADVANCED CARTRIDGE PROTECTION

As explained in Atari Software Protection Techniques Vol I, the field of software protection and copying methods is rapidly changing. The cartridge area is no exception. Several new hardware and software advances have been made since the publishing of Vol I. This chapter will explain the state-of-the-art protection and copy techniques as they are today. It is recommended that you review Chapter 8 of Volume I before reading this section.

Saves to Disk

Cartridge back-ups are usually made by saving the cartridge data to a disk, then running the backup by reloading the data to RAM. One technique that prevents these back-ups from running is to have the program overwrite itself. The original can't overwrite itself, because it's stored in Read Only Memory, but a copy running from RAM is destroyed. Previously, the only way around this protection was to break the code by hand, or copy the cartridge to EPROMs. Now, a new method has arrived.

Pseudo Cartridges

Pseudo Cartridges enable users who have little or no programming knowledge to use a disk file copy of a cartridge program, without changing the disk file in any way. These cartridges disable parts of the computers RAM, the same memory area that is write-protected when the original cartridge is used. The cartridge program cannot write to this area

when the pseudo cartridge is installed, just as it cannot write there when the original cartridge is present. This prevents the program from overlaying or destroying itself.

Some popular commercial examples of pseudo cartridges are the Impersonator, the Pill, and the Super Pill. Each has it's own advantages and disadvantages, and each will allow slightly different things (see The Tools section for reviews).

All pseudo cartridges use the same basic principles and work in the same way. First, the cartridge program is saved to disk as a binary file. If a special menu or boot program is needed, it is added to the disk. When the disk is used, the menu or boot program is loaded, which then loads the cartridge data into its normal area of memory. Then before the cartridge program is run, the back-up program stops for a moment. Then the user can insert the pseudo cartridge (or turn it on, if it is equipped with a switch). The pseudo cartridge protects the RAM so the cartridge program runs normally. In other words, with the pseudocartridge in place, the program can no longer overwrite itself, and will run just like the original cartridge did.

The disk files created by these commercial cartridge back-up systems are usually copyable, so several disk back-ups can be easily made. The disadvantage is that most disk files will not run without the pseudo cartridge installed in the computer. Some of these commercial back-up systems will allow some older, unprotected cartridge programs (Pong, Star Raiders, etc.) to run without the pseudo cartridge. Some of these systems come with special menu programs that must be used with the disk files and the pseudo cartridge. The more expensive systems have an on/off switch on the pseudo cartridge. Once again, the pirates had won another round in the battle of software protection. But software publishers didn't sit still either.

Bank Select Cartridges

How can you make a 40K program run on only 16K of memory? With a new breed of cartridge called a bank select cartridge! Bank select cartridges

were originally developed for the Atari 2600 game machine to help overcome the drawback of its tiny 2K memory. They were introduced on the computer to squeeze bigger programs into smaller areas of memory, but they have one additional advantage; they are extremely difficult to copy.

Bank Select cartridges are physically different from a standard Atari cartridge or EPROM. A bank select cartridge contains not one, but two sets of ROM chips. The Bank Select cartridge can instantly switch the separate memory banks in and out when needed. This requires complicated hardware and programming, but the benefits, especially the extra available memory, are worth it in some cases.

Bank select cartridges present two major problems to a pirate trying to copy them. The first problem is the extra memory. A bank select cartridge holds a lot of data, but takes up only a small amount of memory. A 40K program may take up only 16K when it's stored on a bank select cartridge, but would take up the full 40K if it were read into RAM. The second problem is that the code is written to the configuration of the bank select cartridge. The program on a bank select cartridge is divided into sections that can be called into the same memory area as they are needed, so each section is assembled to run in the same range of addresses. Any program written like this would require extensive modifications to run correctly when the entire program is loaded in RAM.

Because of the high manufacturing costs, bank select cartridges have very limited uses. Currently they are used in several special language cartridges produced by O.S.S. to preserve memory. Only one mass produced game (Bounty Bob), currently uses a bank select cartridge, but as more sophisticated applications requiring more memory are developed, they may become more popular.

Overcoming Bank Select Protection

There are two basic methods of copying Bank Select cartridges. Neither one is very practical or easy. Below is a description of both.

Reverse Engineering/EPROMs

This copy method involves studying the cartridge hardware, analyzing the circuitry and chips, copying the ROMs to EPROMs, then rebuilding a duplicate cartridge from scratch. One problem is that no standards exist to govern how a bank select cartridge should work. Each cartridge may bank in different areas of memory, depending on the requirements of the individual program. In order to create a copy, each bank select cartridge must be studied and handled individually, according to it's own structure. An additional drawback is cost. The cost of buying all the necessary EPROMs and chips, and etching the boards may come close to the price of the original cartridge. If the original is bought from a mail-order house or a discount house, the price of making a bootleg copy can be higher than the price of an original! To most pirates, the time, trouble, and expense of making a duplicate are not worth the effort. Manufacturers have economies of scale, so most individual pirates cannot hope to beat the cost of these cartridges.

Breaking Bank Select Cartridges By Hand

Breaking a bank select cartridge by hand is one of the most difficult tasks a pirate can attempt. Even when a pirate has the skill to break it by hand, the back up copy may have so many drawbacks that it is not worth the pirate's efforts.

The first step in breaking a bank select cartridge is to read the different banks, and store them as a file that can be manipulated as needed. This can be tricky, but it's not impossible. It can be done by removing the chips and reading them with an EPROM burner, or by modifying a cartridge reading program to activate each of the banks and saving them to different files. The next step is to make the file run like the original. Several problems must be overcome to do this.

The bank select hardware loads in each section as it's needed. One alternative is to load each section from disk when it's needed. This is possible, but the time required to load each section makes it impracticable. The original cartridge can switch

banks instantly, and may do so several times for each operation it performs. If a particular operation requires many bank switches, then trying to load each section from disk would make the process unbearably slow.

A more practical, but more difficult, method is to disassemble the separate sections of code, then reassemble them to alternate areas of memory. It's harder than it sounds, because commands that originally pointed to different banks must be redirected to point to the appropriate areas of memory. Even if this reprogramming can be done correctly, a serious drawback still remains; memory usage. A bank select cartridge that requires only 16K when run from the cartridge may require a full 40K of RAM to run properly when run from a disk file. This may be fine for a game program, but renders a language or utility cartridge almost worthless.

Finally, a method that has never been fully tested appears to be a partial solution to the memory problem. This method is to load the banked memory of a 130XE with the separate banks of the bank select cartridge (repeating any common blocks in each bank), then changing the program to toggle the 130XE memory banks, instead of the banks on the cartridge. Of course, this method would only work with a 130XE or equivalent, but may overcome some of the drawbacks above.

Conclusions

The high cost of manufacturing bank select cartridges have limited their use to a small handful of commercial programs. With such few programs, and the inherent difficulties of copying them, pirates have virtually ignored these cartridges. In addition, there have been few regular non-banked cartridges released in the last few years. This is probably due to the high manufacturing cost, memory limitations, and the ease in which they can be copied. Except for language cartridges, which are more functional than disk based languages, and cartridge based educational programs which are easier for young children to use, the era of the computer cartridge program seems to be drawing to a close.

Chapter 7

CASSETTES REVISITED

Vol I discussed the general principles of cassette protection, and how backup copies of cassettes are made. This is a complete discussion of the workings of the progran recorder, and some protection methods.

THE WORKINGS OF A PROGRAM RECORDER

The Recorder

The Atari Program Recorder is really a regular stereo tape recorder with some minor added features. A stereo recorder is different from a regular recorder because it has two separate tracks on which to store information. These tracks are defined as left and right.

The left track is called the audio track. It stores sounds such as voice or music. On an Atari program tape this track is usually blank. When this track is used, the sound it contains is played through the speaker in your TV or monitor.

The right track is called the digital track. It stores information in digital form such as a program or the data a program uses. Although it sounds like noise, the Atari computer can understand the information. It, too, is played through the audio speaker of your TV/monitor.

The information on the digital track is made up of two different sounds. One is called the space and the other is called the mark (3995 Hz and 5327 Hz respectively). These marks and spaces represent bits, and make up the information which the computer

uses. Instead of the usual 8 bits per byte, tape bytes have 10 bits per byte. The first and last bits are used to define the begining and end of the byte. A space indicates the start of a byte, and a mark means stop. The bits are stored on the tape at 600 baud, or 600 bits per every second. They're stored on the tape in groups, each group contains 132 bytes of data, and is called a record.

The Record

The first two bytes in a record are markers for speed measurements. The baud rate of the program recorder is assumed to be 600, but it can vary slightly. So, the speed is calculated using the first two markers in the beginning of each record (55 hex or 01010101 binary). When the first marker is received, POKEY saves the vertical and frame counters. Then, at the end of the second marker, it uses the new vertical and frame counter to compute the actual baud rate. This baud rate may range from 318 to 1407, and is redetermined for each record.

The next byte (the third byte) is the control byte. The control byte can have one of three values. The first value, $FC, indicates that the record is a full record. A full record consists of 128 bytes of data. The second value, $FA, indicates a partial record. The number of data bytes in the record is stored in the 128th byte of the record. The bytes between the last data byte and the 128th byte are present, but empty. The final possibility is $FE, which indicates a end of file record. All records have 128 data bytes following the control byte, even if the record is a partial or end of file record.

After the data bytes comes the checksum byte. This byte is used to check to see if the data bytes have loaded correctly. As each byte is loaded, it is added to a partial sum. If a carry byte is present, it is added too. If the record loaded correctly, the final value of the partial sum equals the checksum byte.

The File

The first part of the file is the leader. When the program recorder is opened for output, the motor starts, and a 20 second 'mark tone' is recorded on the tape. Then the Operating System returns to the user, but it continues to write marks for 35 seconds, when a timeout occurs.

The rest of the file consists of records. These records start with a Pre-Record Write Tone (PRWT), then 132 bytes (2 speed marker bytes, 1 control byte, 128 data bytes, and 1 checksum byte), and finally a Post-Record Gap (PRG). The PRG of the first record and the PRWT of the second make up the Inter-record Gap (IRQ). The IRQ consists of mark tones. The length of the tones depends on the IRQ mode: Normal or Short.

In the Normal IRQ mode, the computer stops the program recorder between records, giving the CPU time to process the data. The PRWT is 3 seconds. The PRG is up to 1 second of unknown tones.

In the Short IRQ mode, the program recorder never stops. The PRWT time is .25 seconds. The PRG time can be any length up to a user specified limit.

The Cassette Boot

The program recorder can be booted only during power-up, but it doesn't have to be booted to be used. You can boot the recorder by holding down the 'START' key during power-up. After you hear an audible signal, pressing any key will load a machine language boot program from the program recorder.

A boot file is defined by the first six bytes in the first record. The first byte is unused. The second byte contains the number of records in the boot file. The third and forth bytes contain the low and high bytes for the memory addresses where the rest of the file is to be stored. The fifth and sixth bytes contain the low and high bytes for the memory address where the operating system will jump to after it has loaded the file.

Loading Problems

A common problem with the Atari Program Recorder is its inability to load and save information with 100% accuracy. Failures happen mainly because it is not a high quality product, and the user doesn't use high quality tape.

If the recorder is not loading tapes made earlier on that same recorder, chances are the problem is the quality of the tape. The best tape for an Atari is a high quality audio or digital tape in a short length. Discount tapes tend to lose information. Long tapes tend to be too thin, and sometimes they stretch easily. Five to fifteen minute digital tapes or 30 minute Maxell or TDK tapes are the best.

If your recorder is not loading tapes made on a different recorder, the problem is most likely tape alignment. This is happens when the two tape heads read and write at two different locations. This problem can be fixed by turning the head adjust screw, But beware - the tapes you have made previously on your recorder may not load. To solve this dilemma, first adjust your tape head so the recorder can read the tape, load the tape, then change the head back to its original position, and resave the file. An alternative to adjusting the tape head is to use both recorders with the same computer. Load the tape with one recorder, and save it with the other. Using a program like Cassette Operating System (from Alpha Systems) will make this method faster and easier.

Adjusting the tape head is not difficult, but it can be very tricky. Be careful if you try it. The head position is controlled by a single screw. To find it, open the recorder cover. If there's a tape inside, remove it. Look for the screw on the left side of the read/write head. Mark the original position of the screw and count number of turns, so it can be set back to the original alignment. To adjust the head position, turn the computer on, with BASIC loaded in. Put a tape under the cover (not in!) and press 'PLAY'. Then POKE 54018,52. This poke turns on the recorder's motor. Turn up the TV/monitor volume until you can hear what sounds like the

computer loading a file. Now turn the screw with a Phillips screwdriver a little bit at a time. When the sound is the loudest it's properly adjusted and aligned. To adjust the recorder's alignment to match the alignment of another recorder, turn the same scew, and experiment with different head positions, until you find the one that works. Keep track of how far the screw is moved, so it can be returned to it's original position.

Copying Files

Copying a cassette file is easy. LISTING 1 is a file copying program. The program uses a file routine to set up an ordinary input/output control block for the cassette to read or write. It then jumps into the operating system at $E456 to CIO. The operating system will load or save the file.

Copying cassette files to disk files can pose some problems. A major problem is Atari DOS. Most cassette files usually occupy the same memory. Multi-load cassettes, or cassette programs that load additional files from the cassette recorder also cause problems. Utility programs such as COS (Alpha Systems) are designed to overcome these problems.

PROTECTION TECHNIQUES

The most common cassette protection technique is the double file boot. One file is booted, then the first file boots a second file. This technique is easily copied. Load the first file, then load the second file. After that, save the first file, then the second.

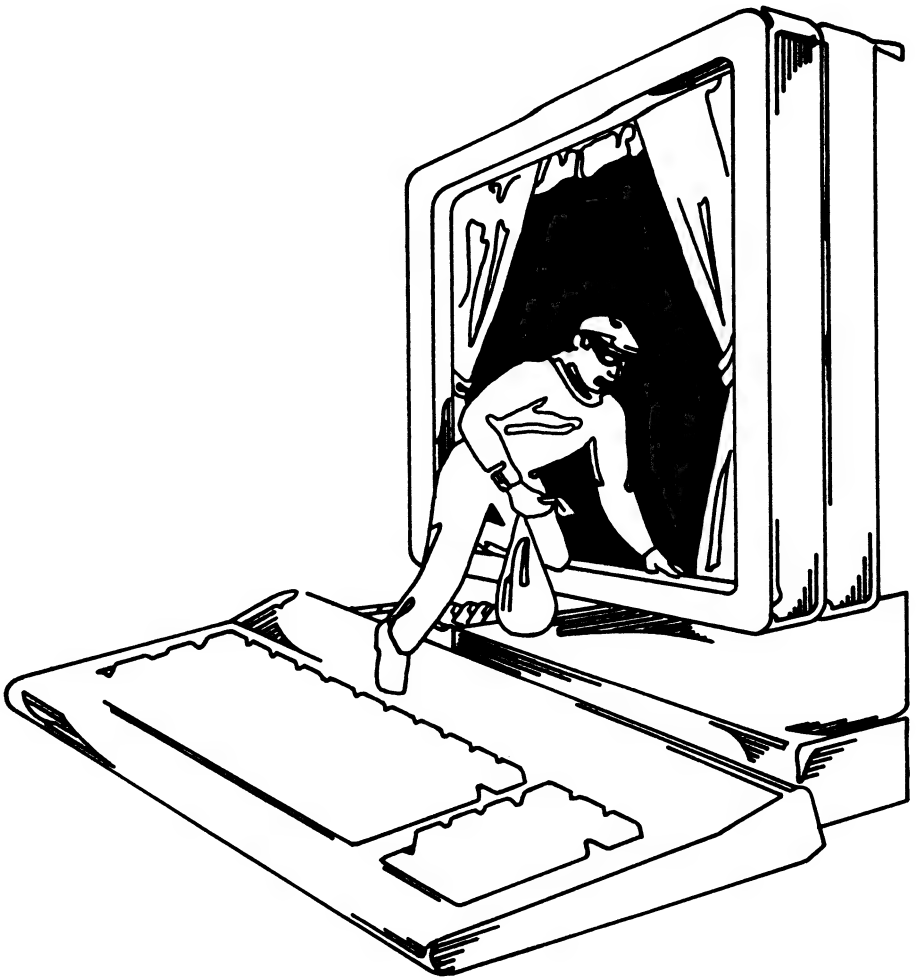
One of the hardest techniques to copy is the short leader technique. This technique uses a second file with a shorter leader than the operating system normally allows. It's impossible to load this file using the operating system's load routines. A program which acts as the operating system, but requiring a shorter leader time is needed to load this kind of cassette.

No matter how complex or devious the copy protection scheme is, a cassette can always be copied with a good quality double audio tape deck.

Cassette File Copier

```
10 REM CASSETTE FILE COPIER
20 REM BY George J Polly
30 REM
40 DIM ML$(40)
50 REM SAFE = END ADDR OF PROGRAM
60 SAFE=PEEK(144)+PEEK(145)*256+100
90 REM PUT ML ROUTINE INTO ML$
100 FOR I=1 TO 30:READ A:ML$(I,I)=CHR$(A):NEXT
I
110 DATA
104,162,16,169,7,157,66,3,104,157,69,3,104,157,68,3,104,
157,73,3,104,157,72,3,32,86,228,132,208,96
120 REM
130 REM START MAIN PROGRAM
140 REM
150 REM *** LOAD SECTION ***
160 ? "Push PLAY.":? :? "Press RETURN to load
your file.":?
170 REM SET LENGTH AND OPERATION
180 REM JUMP TO FILE ROUTINE
190 IO=4:LENT=40000:GOSUB 1000
200 REM CHECK IF EOF ERROR
210 IF A<>136 THEN ? "LOADING ERROR!!!":END
220 REM FIND LENGTH OF FILE
230 LENT=PEEK(856)+PEEK(857)*256
240 REM *** SAVE SECTION ***
250 ? "Push PLAY and RECORD.":? :POKE 764,255:?
"Press RETURN to save your program.":?
260 REM SET LENGTH AND OPERATION
270 IO=8:GOSUB 1000
280 REM CHECK IF OPERATION COMPLETE
290 IF A<>1 THEN ? "SAVING ERROR!!!"
300 REM DONE WITH COPY
310 ? "DONE.":END
1000 REM
1010 REM *** LOADS OR SAVES FILE ***
1020 REM
1030 REM LENT = LENGTH
1040 REM IO = IO OPERATION
1050 REM
1060 REM SET ML ROUTINE FOR OPERATION
1070 ML$(5,5)=CHR$(IO+3)
```

```
1080 REM OPEN FILE
1090 OPEN #1,IO,128,"C:"
1100 REM DO OPERATION
1110 A=USR(ADR(ML$),SAFE,LENT)
1120 REM CLOSE FILE
1130 CLOSE #1
1140 REM SEND POSSIBLE ERROR BACK
1150 A=PEEK(208)
1160 RETURN
```



Chapter 8

ON-LINE PROTECTION

The field of protecting on-line systems from misuse has become one of the most critical and talked about today. While protecting a program from piracy may seem important, on-line protection can take on a global siginificance. Today computers control everything from automated traffic control systems, satellite control systems, and even our national defense network. The American Bar Association placed the losses to businesses at between 145 and 730 million. Press reports on hackers and movies such as War Games and Superman III have brought the problems of on-line protection to the forefront. Much was said earlier in the book about who hackers are and the way they operate. This section will show what can be done to help stop it, whether you wish to protect a national defense network or your personal bulletin board system.

A recent survey of data processing managers showed that 35% of the installations had suffered security breaches involving unauthorized use of the system. Of those who had the problems, though, only 8% of all security breaches and only 6% of actual computer crimes were committed by outsiders. All the rest were due to employees. Also, of the employees, only 30% were workers in the computer department. So it seems that the majority of computer crime is done by employees spread throughout the company, as opposed to outside hackers or computer wizards working within the company.

GAINING ACCESS

Back Doors

One of the most common methods used to gain access to on-line systems is through back doors. Back doors are special accounts set up by computer manufacturers, repair people, and programmers to allow them special access to the system. These accounts are almost never listed on the company's list of valid accounts and are usually known about just by the people who set them up. The power of these back doors was shown recently after the discovery of a hacker in the Seattle area.

An 18 year old computer wiz, Michael Princeton Wilkerson, recently hacked into several major computer installations. He changed data and even set up a code bomb that could have destroyed the whole system (see Logic Bombs and Program Worms). He did so using one of the most widely exploited back door attacks; using the special service accounts and default passwords set up on all VAX computers when they are installed. The accounts (SYSTEST and FIELD) are supposed to be used by field engineers and service people during installation, but then should be changed. These standard accounts have been listed in hacker bulletin boards and hacker newsletters like 2600 Magazine (see Hacking), and even the National Bureau of Standards issued a warning about them back in 1984. It was these same passwords that the famous "414 Gang" had used to penetrate Lawrence Livermore National Labs several years ago. But as Michael proved, these back door accounts still work in almost every installation he tried.

There are many other examples of back doors, including one on many Atari bulletin boards. Most versions of the popular 'Forum' bulletin board system for Atari computers have a back door. The back door was probably placed there by the original author, Matt Singer, to watch for people who might misuse his program. It allows a caller who logs on as 'Matt Singer' to issue special system commands that let him act as a remote sysop, reading,

deleting, or changing even the system files at will. Although this particular back door has been found and removed from some versions, many people who have modified or improved the system have added their own.

The widespread use of back doors undermines any serious attempt at on-line security.

Passwords

Almost all secured systems require the use of a password. Most systems allow the user to set up and change his password as desired, and that is a big part of the problem. Even at large security conscious corporations, many faulty practices occur. Passwords that are one character in length (obviously simple to crack) are used. Whole groups of users sometimes share a common password. Passwords may be posted by the terminal or scribbled on the desk in case people forget. Another practice is to choose a password and never change it. All of the actions stand in the way of good security and are easy to prevent.

Users should be forced to enter a unique password (not used by someone else) of 5 characters or more, then forced to change it once a month. The new password should be compared against the prior passwords to be sure the user doesn't just change it back (a common practice) or switch back and forth between two. Finally the users should be educated not to post or give out their password.

Phony Log On Trick

The phony log on trick is a method some hackers use to trick other users into giving up their passwords. One noted case occurred on Comp-U-Serve. The message "-SYSTEM COMM ERROR- Please Logon" was sent to unsuspecting users on the CB simulator. When they responded, they were prompted to enter their IDs and passwords which the perpetrator later used to access Comp-U-Serve and enter their accounts.

A more common practice is for a hacker, after breaking into a new system, to insert a special program in the log-on sequence (called The

LOGPROC). The program makes it appear to the users that he didn't get on and it prompts him to re-enter his ID and password. These are stored in a file for the hacker who, upon returning to the system can have built up hundreds of accounts. One method to prevent this from working is described below.

Automatic Call Back

This relatively new system has been successfully used to thwart outside hackers at many companies. When the user calls the system and gives his ID, the system immediately hangs up and calls the user back on the number stored for that ID. Only then does it prompt the user for his password and allow him to log on. This makes it almost impossible for a would be hacker to get on the system. He not only needs the ID and password, but he must be at the registered users phone number upon call back as well. This is one strong way to be sure the user is who he says he is.

New High Tech Solutions

A number of new methods to prevent unauthorized access have recently been implemented. They include terminals that require special magnetically coded cards, that can read the vein pattern in the eye and even one that can check the users finger prints. These gadgets may be good for protecting the national defense, but are usually too expensive and cumbersome to be put in all terminals. One solution that can used on any terminal is called the smart card. This is a computerized clock the size and shape of a credit card that displays a different code every 60 seconds based on a formula using the time and the user's ID number. The computer asks for that number when the user logs on and verifies it by applying the same formula. This password that changes every 60 seconds almost guarantees that the user has his card. In another "smart card" variation, the user is given a card that looks like a calculator. Then when he logs on, he is given a number which is typed in

on his card. The card calculates and displays a response number which the user types into the terminal for access. These and other new high tech methods may go a long way to help prevent the hacker problem.

MISUSING THE SYSTEM/PROTECTING YOUR SYSTEM

This section discusses some of the ways hackers may misuse a system and some additional things that may prevent them. This information can be used by everyone from a small bulletin board sysop all the way up to the manager of a large data center.

The HELP Command/User Friendly Systems

Obviously most people want to create an easy to use system that offers a user help whenever it is needed. Unfortunately, crackers can use and misuse this information to learn your system. Almost all systems have a HELP command that gives at least some information about using the system. A system operator must strike a balance between an easy to use system and a difficult to crack system. Obviously, a system that requires no password, ID, etc. is easier to log onto, but is also much less secure.

The trick to writing good HELP screens is to instruct users how to do passive acts, such as reading files, but not reveal information on topics like deleting system files or formatting drives. You could assume that anyone who is sophisticated enough to properly use those commands would not need the simple HELP supplied on the menu. Whether this assumption is right or not, it can prevent some problems.

Access Levels

An access level on a system determines which commands, activities, and files a specific user is allowed to access. Limiting users to only the functions that they are supposed to use helps to secure a system in two ways. First, it prevents a

legitimate user from accidentally doing things he shouldn't, and second, it can help keep a hacker from gaining complete system access.

Access levels should be set up so that each user can do only what he needs to do. Frequently, a level may specify that a user has read-only access to all files except the ones he creates himself. He can be further restricted to be able to read only certain files and use only certain commands.

Security Packages

For years, there have been several commercial packages available to help protect mainframe systems (such as RACF from IBM, and Top Secrets from CGA computer), but small sysops usually have only what was provided with their BBS system, or whatever they can do on their own. This situation is now changing. With micro computer use growing stronger in the business community, several packages have been released for the IBM PC to protect data. Systems such as Mailsafe from RSA Data Security Inc. and Pro-Tek from First Byte allow users to protect their files by using data encryption, passwords, or special formats. Atari users can use the programs on the disk included with this package to do some of these same functions.

Data Protection/Control Codes, Backups, Etc.

No matter how secure you feel your system may be, there are certain precautions that should always be taken. Below are a few which should be considered.

1. Always write protect your system disk to prevent accidental or intentional destruction or modification of your password files, access levels, etc. Any new information or changes should be added by the sysop after it has been reviewed.

2. A BBS program should watch for control codes that are sent to devices. For example, a popular way to crash some Forum boards is to leave a message for the sysop, (which prints on the printer) containing control codes to de-select the

printer. Some modems also allow themselves to be controlled from a remote system, but this feature can usually be turned off with the proper control codes.

3. Always have backup copies of all data accessible on line. This not only can save you from a hacker, but is necessary insurance in case a disk fails on its own.

STOPPING INSIDERS

As stated in the opening of this section, most damage is done not by outside hackers, but by insiders who know the ropes of the system. Even the best security measures become ineffective against the insider because he may know the security well enough to circumvent it. The real key to security against inside jobs is effective personnel management. Be sure the people given access are trustworthy. Aside from that obvious step, there are several other things that can be done. The August 19, 1985 issue of INFORMATION WEEK magazine presented an extensive discussion of this subject. Below are some of the major conclusions they reached.

- Split responsibility for the system among several data processing staff members so that no one has total access.

- Change passwords regularly.

- When an employee leaves the company, eliminate his account immediately.

The Technical Approach

- Implement data encryption on personal computers. This helps prevent local compromise of data stored on PCs and LANs.

- Add front-end processors for another layer of security to prevent unauthorized

access from external sources.

- Maintain an audit trail of information transferred to microcomputers.

- Restrict access to proprietary data by segregating it on a separate system.

- Restrict unauthorized access to personal computers; keep PCs out of public areas; add power locks to PCs with hard disks.

- Require positive identification of users beyond simply names and passwords. Card access or biometric identification (fingerprints, handprints, or the like) serve well here.

The Managerial Approach

- Establish a data security policy.

- Establish procedures to implement the security policy, including a system for classifiying data and preparing for disaster recovery.

- Include adherance to the policy and procedures as a part of employees' responsibilities and a component of employee evaluation.

- Screen prospective and current employees before granting access to computer data.

- Limit the amount of information to which any one employee has access.

Chapter 9

THE LAW

Piracy has continued to be a growing problem in spite of the fact that duplication and distribution of copyrighted software without the copyright owners permission is illegal. Part of the problem stems from the age of the laws and the penalties they invoke. The United States copyright laws are over two hundred years old, and the concepts behind them are even older, descending from English Common Law. Most of the Copyright laws were originally geared towards printed materials, which are harder to duplicate than software. This chapter will discuss the laws, both current and proposed, regarding copyrights and software.

LEGAL PROTECTION METHODS

A company which tries to use the law to prevent software piracy is said to be using a legal protection method. Software piracy is illegal. Unfortunately, legal tactics alone are ineffective. To understand why, first we'll take a look at how these forms of legal protection work. First, this chapter will discuss software licensing and what rights a software buyer has. After copyrights and patents, this chapter will discuss the new trends in software law, including criminal convictions for copyright violations, crackdowns on pirates, and new data security laws.

The Uniform Commercial Code and Software Licensing

The Uniform Commercial Code (applicable in most states) is the section of law governing

commercial business transactions. This is the body of law that covers almost every transaction in the day to day economic life of this country. But just how the UCC applies to computer products is a grey area. The UCC does not cover services like computer programming, but, when the programing is "bundled" with hardware (hardware and software sold together as a package), it may become applicable. In general, computer hardware is cover by the UCC, but software falls outside of it. The reason is that most software is considered to be licensed rather than sold. One exception to this rule is software sold with hardware, then the whole package is covered by the UCC.

Software Licensing

Most legal officials feel that computer software is outside the Uniform Commercial Code, because a software buyer is not really buying the product, he is just licensing it. Many software packages carry a disclaimer on the wrapping which states that the buyer is not purchasing all the rights to the software, only the right to use it. In other words, what is being sold is a copy of the software for use by the buyer only, not the software itself. The other rights that are not sold are called property rights, or rights that come with ownership of property. In this case, the property is the software, and the legal owner of the property is the software publisher. The buyer owns only permission to use the copy of the software that he purchased. This sale of limited rights to the software is called licensing. Illegal distribution of the software is a copyright infringement. Copyrights are discussed in Vol I and later on in this chapter.

Some large companies and schools need many copies of a single program, often for different systems and stored on different media. A software publisher may agree to sell such a company a number of copies of a program, and for an additional fee, sell the company the right to produce other copies for that company's use only. This type of agreement is called site-licensing, and is discussed in full later in this chapter.

Trade Secrets and Copyrights

Vol I explained what trade secrets and copyrights are, the role they play in software protection, and the advantages and disadvantages of each. Since that time, many new and interesting developments in software copyrights have occurred. Copyrights have continued, and will continue to be the primary means of legal software protection, so first we'll discuss copyrights and copyright registration, including how to register a software program. Patents were also explained in Vol, I, and new developments in this area will also be covered.

How To Register For A Copyright

Originally, an application had to be filled out before a copyright could be granted, but now a copyright is automatic as soon as the work is completed. Without formal registration however, it may be difficult to prove that one work was finished before another.

The Copyright Office classifies different kinds of works (books, records, plays, etc.) into different categories. Computer programs are classified as machine-readable non-dramatic literary works. The requirements for each classification vary, so only the registration requirements for computer software are discussed in this book. Information on obtaining registration for other kinds of works are available from the Copyright Office.

The registration process is simple, but the Copyright Office is a bit slow, so it can take over three months to receive a certificate of registration. The first step to to fill out the appropriate form, for computer software, it is Form TX (see the end of this section for complete information on obtaining this form). It's a fairly straight forward form, it asks for the title(s), the author(s), names of any other copyright claimants, a section to complete if the form is an update to an existing registration, who will manufacture the work, and if the work may be reproduced by the Library of Congress for the blind or deaf.

The title is, of course, the name of the work.

While the author is usually the person who actually wrote the work, one exception to this is when a person writes the work as a part of his job. Then the work is said to have been "made for hire". In this case, the company that the writer works for is considered to be the author. If the author sells his work and the rights to it, the buyers are the copyright claimants. If the author never sold the rights, then he is the copyright claimant.

If this is a re-registration, the Copyright Office will ask for the previous registration number and the date, and any information on "derivative works". Derivative works are works which are made from another work. For example, a novel might be an original work, and a version of the novel condensed into a short story might be the derivative work.

The Copyright Office will need to know who is manufacturing the work (who is making the copies for distribution, such as who is printing the books, who is duplicating the disks) because works that are manufactured outside of the U.S. and Canada are not fully protected. The Library of Congress reproduces works in forms that can be used by the blind and deaf. They reproduce works only with the permission of the copyright owner. You do not need to grant this permission to register or receive full protection under the law. If you grant permission, and later change your mind, you may cancel this permission upon 90 days notice.

The fee for copyright registration is $10.00, and must be returned with the application form. Additionally, you must submit a deposit of the work. For most works, two complete, readable copies are required, but Section 202.20(c)(2)(vii) of the Copyright regulations states that for machine-readable works (software), the entire work need not be submitted in visual form. This means that you do not have to send a complete listing of your program, only a partial listing. In most cases, this is the first and last 25 pages of the program.

The effective date of registration is the day on which the Copyright Office receives the registration form, the filing fee, and the deposit copy. The Copyright Office will not acknowledge when it has

received the material, and the actual certificate may not be issued until three months later, so it's a good idea to send the materials certified mail return receipt requested. The return receipt is acceptable proof of the registration date. The address is:

Register of Copyrights
Copyright Office, Library of Congress
Washington, D.C. 20559

All works published with the consent of the copyright owner need to be identified as copyrighted. If copies without the copyright notice are published, the copyright owner may lose some rights. The copyright notice is the copyright symbol, c , the year the work was first published, and the name of the copyright owner.

All the necessary forms, and additional information can be obtained from the Copyright Office. You can get the forms by calling (202)-287-9100, or writing to the Copyright Office. The address is:

Information and Publications Section LM-455
Copyright Office, Library of Congress
Washington, D.C. 20559

Registration: Pros & Cons

Registration has some advantages, but it is not required to receive full protection under the law. Registration can be used to prove that one program was completed prior to another program. A program must be registered in order for a publisher to file suit to claim copyright infringement.

One drawback to registration is that it may invalidate any trade secret protection. When a copy is given to the copyright office, it becomes public information. Anyone can see your coding, so the material is no longer confidential. If your program source code is longer than 50 pages (we've never done it), you can avoid this problem by registering only the first and last 25 pages of code. As long as the crucial parts are not in that section they will remain confidential, insuring trade secret protection also.

The Copyright Owners Rights, Fair Use, and Penalties for Infringement

The copyright owner has exclusive rights to reproduce the work in copies, to prepare derivated works (works derived from the original), to distribute the work, and to perform or display the work publicly. Source code has always been copyrightable, and the Computer Software Protection Act of 1980 explicitly made object code and ROM copyrightable also.

The one exception to a copyright is the Fair Use Exception. This exception allows limited copies for personal use only. You can make a photostat of a recipe from a cookbook and keep it in a recipe file, or make a back-up copy of your program for safekeeping, but you cannot distribute those copies to others without permission. Teachers, for example, cannot make Xerox copies for an entire class, although many do anyway. This is especially important in software, because it means that a teacher cannot make a copy of a program for each student without permission from the publisher.

You can sell your copy of the program, but you cannot change the contract made when you purchased the software. For example, if you buy a book from a bookstore, you can do whatever you like with that copy of the book. You can read it and give it away, or you can sell it to someone else. But you cannot sell the right to print and distribute other copies, because you never bought that right. You can buy a copy of a program, and then sell that copy to someone else, but it is illegal to make several copies of your copy and sell them all. The penalties for copyright infringement include injunction, imprisonment, and fines of up to $50,000 per illegal copy.

Patents

Patents are another form of legal protection. Getting a patent is an expensive and time-consuming process. It takes at least a year from the date of application, and can cost thousands of dollars. A patent protects the idea as well as the expression, so protection is much broader. To receive a patent,

the invention must be directed toward statutory subject matter. Statutory subject matter is something that can be expressed in concrete terms or written form. An example might be a design for a chip (subject matter), expressed in a blueprint (concrete terms). You may want to refer to Vol I for more information on patents.

In the early 1960's, many programs were patented. In 1965, a presidential commission recommended against patent protection for software, so no software patents were issued for the next 15 years. The commission felt that because the ideas were expressed in mathematical formulas and programs, and used on computers, they were not statutory. In 1981, the Supreme Court ruled that software was, indeed, both statutory and patentable.

The Supreme Court's ruling did not unleash a flood of new patents. The process is long and costly, and even confidential parts must be disclosed to the patent office, which invalidates any trade secret protection.

A few software companies have decided that the time and trouble may be worth the protection that a patent offers. In most cases, the companies have chosen to patent only a portion of their programs. Businessoft obtained a patent in September, 1985 on part of a program called Mindreader. It's a word processing program that includes a word completion routine. The routine is the only part that was patented.

Decision Support Software is another firm which has chosen the patent approach. In March 1986, the U.S. Patent office approved an application for the patent of the operation and screen display for Decision's Expert Choice system. Mary Ann Selly, the firms president, says "We want it [the screen display and operation] to be unique and associated only with Decision Support Software."

Quickview Systems, Inc, has spent over $20,000 pursuing patents for it's software. After two years of waiting, they were recently awarded a patent on one of their products, a text compression package called Zoom Racks. President Paul Heckel feeis that Quickview was lucky that it's patent was awarded so

quickly. He says the idea for Zoom Racks is so unique and complex that nobody will be able to duplicate it for a few years anyway, "But then they'll have to do business with us or take the chance of a patent battle".

Patents will continue to play a role in software protection, but because of the time, expense, and restrictions they will never replace copyrights as the primary legal protection. Companies will patent important sections of programs, such as screen displays and difficult or unique routines, because patents afford greater legal protection.

NEW TRENDS IN SOFTWARE LAW

Untied States copyright laws were established 200 years ago, long before most people even dreamed of computers and programs. Only recently have the laws begun expanding in new directions to protect software programmers in the personal computer explosion. In the past, copyright violations were determined by the concept of substantial similarity. If a copyrighted work and a second work were found to be substantially similar, the second was considered to be in violation of the copyright law. Courts have held that copying source code in its entirety is an infringement on the copyright law.

One of the most significant changes in the protection of patents was the creation of a patent appeals court in Washington D.C. in 1982. The new court is headed by a former patent attorney, and has upheld more than half of the patent suits it has heard.

Fighting a patent or copyright violation isn't always easy. Apple Computer has spent over $5 million on litigation costs since 1981, pressing various foreign and domestic patent and copyright lawsuits, occasionally meeting with success, as in the case against the now bankrupt Franklin Computer Corp.

The costs in fighting these violations run high, because most of the illegal products are produced overseas. Senior vice president and general counsel with Apple Computer Corp., Albert Eisenstat, says

60% of all counterfeit products come from Taiwan, with Hong Kong and Singapore are close behind. While lawsuits in these countries may force the governments to take some action, the violators rarely face serious penalties. In one case, Eisenstat says several executives from an unidentified Taiwanese firm producing counterfeit Apple IIs avoided a 6 month jail sentence "By paying the government 50 cents a day to stay out of the can".

Some parties feel that the solution is for the U.S. to sign the Berne Copyright Convention (the U.S. is the only developed nation that hasn't). It would provide more protection than the Universal Copyright Convention, which the U.S. now follows.

But the situation in the United States is not so bleak. In 1985, a federal court in Pennsylvania considerably broadened the protection afforded a copyrighted program by ruling that copyright infringement exists even though the second program is not identical to the copyrighted work. Attorney Peter Brown says this decision "now clarifies that 'translations' of a computer program, even if they are not literal,...[constitute] infringement." This means that adapting or modifying a program and selling it without the copyright owners permission can constitute a copyright violation. Mr. Brown says "If I start with someone else's computer program and then change it, when does it become mine?...The answer [is] Never." A federal court in Tennessee re-enforced this concept in a similar ruling two months later. The courts were not lenient with the offenders, the remedies for partial duplication of material included loss of profits and injunctions against further work on the products in question.

First Criminal Conviction for Software Copyright Violation

In August 1985, the first Federal conviction involving criminal copyright infringement was handed down by a Federal jury in San Jose, CA. Taiwanese businessman Teh Yi 'Danny' Huang was found guilty of three federal counts of conspiracy, smuggling, and false statements, and two misdemeanor counts of conspiracy to violate copyrights and criminal copyright infringement. Penalties could be a maximum 14 year sentence, and $75,000 in fines. All previous convictions in connection with computer copyrights have been civil cases. This gives prosecutors a powerful new tool to use against some software profitiers.

Sting Operations on Pirates

Another tactic to fight piracy is enforcement of the current laws. Software publishers and the Federal government are continuing the crackdown on sales of illegally copied software. In October 1985, the FBI closed down two alleged pirates conducting business as Lowery Communication and Computer Software Consultants. In December 1985, the FBI raided three alleged software counterfeiting operations. In the third raid, against a pirate doing business as Joseph Duval Co., agents confiscated over 1,000 disks and photocopied manuals, and 10 counterfeit IBM and Apple computers.

The FBI began it's investigation after several software publishers complained that the low-cost computers and programs Joseph Duval Co. was advertising in the classified ads of the Los Angeles Times and other local papers were illegal copies. Lowery Communications and Computer Software Consultants had also been advertising in the classified section of the same papers.

Joseph Armstrong, vice president of finance at State of the Art, Inc, said they had purchased two $595 State of the Art Accounting Software modules from Duval for $50.00 each. Armstrong said his company had also received numerous phone calls from consumers who said they had bought

illegitimate software from the pirates. FBI agents purchased several Apple versions of the pirated programs as a part of it's sting operation. It is believed that Duval's sales were between $2,400 and $3,200 a month.

Investigators suspect that Duval hired kids to break the programs. He was also alleged to have supplied illegal merchandise to another alleged pirate operation run by his sister in Oregon.

Police have also set up sting bulletin boards to catch hackers, these are discussed in the section "Cracking Down on Sysops".

New Data Security, Communications, and Computer Fraud Laws

Software law is an exciting, rapidly changing area. In 1984, congress made it a crime to tamper with a government computer. The state of Ohio is currently considering a computer crime bill.

Legislation to protect computer communications and the rights of private citizens has been proposed in both California and the U.S. Senate. Patrick Leahy's (D-VT) Senate proposal is an update to the federal wiretap laws of 1968. The bill would expand the existing laws to cover digital communication, create standards involving access to computer information by law enforcement officials when probable cause exists, and establish criminal and civil penalties for breaking into private electronic communications systems. The California proposal sponsored by Gwen Moore, ACA 9, is an amendment to the state's constitution to expand protection of free speech and protection against illegal search and seizure to include computer communications.

Some Californian law enforcement officials dislike the amendment. The deputy district attorney for Los Angeles County and head of the department's electronic crime section, Cliff Garrot, feels that the bill may give individuals a license to commit crime, because it requires law enforcement officials to explicitly name the item they wish to search for. But most reactions to both proposals have been positive. The American Civil Liberties Union endorses both. The Electronic Mail Association

Union endorses both. The Electronic Mail Association strongly supports the federal proposal. The San Diego Computer Society, the California Library Association, and El Dorado Teleguide (a public videotext vendor) support the California amendment. Robert Jacobson, consultant to Moore, feels that law enforcement officials have too much freedom now, and "end up looking through everything".

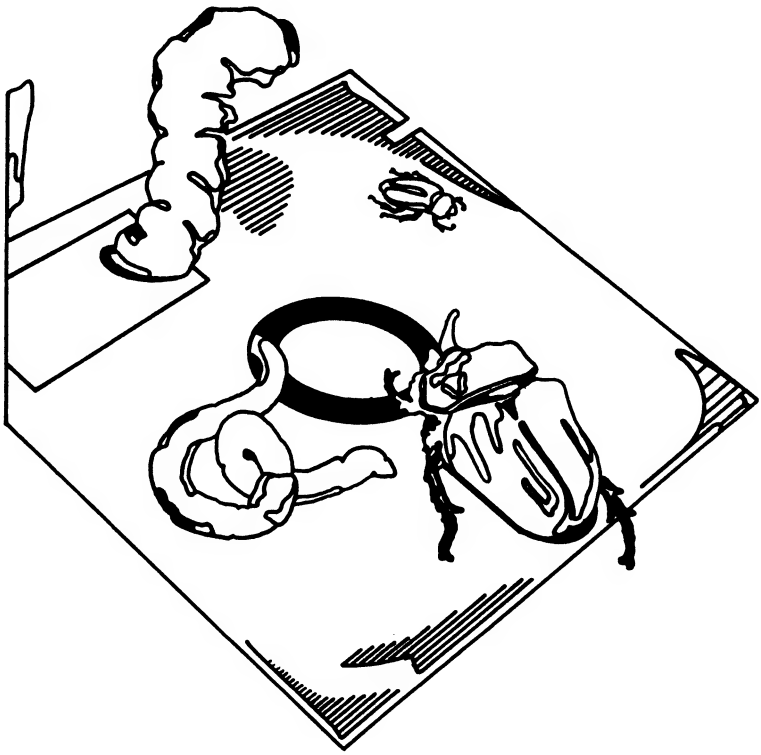
Congress is also considering three other computer fraud bills. Currently, the Comprehensive Crime Control Act states that gaining unauthorized access to classified government data stored in a computer is a felony. Gaining unauthorized access to any government computer is a misdemeanor. Unauthorized access to commercial computers is a misdemeanor only if the computer contains information protected by the Right to Privacy Act or the Fair Credit Reporting Act.

One bill would make it a misdemeanor to gain unauthorized access to any computer system used in interstate or foreign commerce. The second bill would make any computer related crime a misdemeanor if that crime caused the computers rightful owners a loss of more than $5000.00 annually, or if the unauthorized user gained more than $5000.00 annually. The third bill before congress, the Computer System Protection Act, would make computer fraud involving systems used in interstate commerce or federally insured financial systems a federal offense.

Conclusion

Computer software copyrights and data security and protection are legal gray areas. Because the issue of software piracy is so new, there are few statutes and legal precidents concerning it. The confusion arises because software does not exactly fit into the old laws governing non-computer copyrights and privileges. Law officials disagree over what privileges data communications are eligible to receive, including privacy and searches. Judges are begining to decide on cases involving piracy, and these decisions are the beginnings of legal precidents. As problems arise, legislators are writing

new laws to cope with the ambiguity of the old ones. More statutes will be written as the law slowly catches up with technology. If you would like more information on specific laws, see The American Standard Handbook of Software Business Laws, written and published by attorney John Lautsch, a partner in the law firm of Day and Lautsch in Newport Beach, CA, and Chairman of the Computer Law Division of the American Bar Association's section on science and technology.



Chapter 10

OTHER PROTECTION METHODS

DATA ENCRYPTION

Data encryption describes the process of scrambling a file so it cannot be easily read or changed. It can also be used to password protect a file so that the proper decryption key is needed to access, run, or change the file. Data encryption methods vary in complexity from a simple transform table to the complex algorithms used to encode financial transactions.

One of the simplest data encryption methods is to Exclusive-Or each byte in a file by a key byte, then use the same key later to decode the file. An Exclusive-Or is a computer operation (the Assembly language instruction is XOR) which compares numbers one bit at a time. If one and only one of the two bits being compared is a one, then the result is a one. If both the bits being compared are zeros or ones, then the result is a zero. This method is so widely used because the encrypted data can be easily decoded by Exclusive-Oring it again with the same key. In other words, you can encrypt and decrypt data using the same process and the same key. An example is on the next page.

```
0110001011100   Original Number in Binary
0101010101010   A Simple Key
---------------
0011011110110   The XORed Result (Encrypted
                Data)

0011011110110   XORing the result
0101010101010   With the same key takes
---------------  you back to the
0110001011100   original number.
```

Data encryption is sometimes used in programs to make them almost impossible to change. This helps hide any protection code and also helps to prevent pirates from changing things like the authors name and copyright information.

Data encryption can also be used in a data file to keep the data private, or to limit its use to your own program. Strip Poker for the Atari uses this simple data encryption method to protect the picture files from use by others, and many companies are using it for software protection.

At the other end of the spectrum is a process called the Data Encryption Standard (DES). DES is the most widely used encryption scheme for sensitive information such as automatic teller machine transactions. Banks are currently using DES to encode funds transfers totalling 2 trillion dollars each day. DES is also being used in systems such as VideoCypher, which scrambles satellite transmission for companies like HBO and Cinemax. The DES algorithm, developed in the 1970's by IBM, is on the State Department's list of sensitive technologies, and therefore cannot be used on equipment outside the U.S. and Canada.

DES works by breaking up the data into blocks of 64 bits each. First the left and the right 32 bit halves are swapped, then the left half is encrypted with a 56 bit key. This complex logic operation generates another 32 bit packet, which is XORed with the right half 32 bit packet. This new 32 bit number is used to replace the right half of the original 64 bit package, then the entire process is

repeated another 15 times to yield the result.

Although it sounds time consuming, special chips have been developed which perform the calculations almost instantly. Sometimes data is further encrypted, producing a layered effect, with one technique encrypting the result of the previous. This daisy chain of encrypted keys is extremely secure, and is finding more and more uses as the need for privacy increases.

There are many other encryption methods gaining favor for special applications. One, the Public/Private key can be used to secure messages without telling the recipient your encoding key. By using the sum of large prime numbers, you can encode a message so that only the proper recipient can decode it, while still keeping his key, and your key, private. This method is popular for private E-Mail systems.

Although high powered computers can usually crack these schemes given enough time, these methods have been found to be secure enough for almost any use. The software disk included with this package contains two encryption programs. Both encrypt the file with a 64 bit algorithm. They are useful for stopping disassemblers, keeping your title screens intact, and optionally, allowing you to ask for the key before the program will run.

SITE-LICENSING

Companies with large computer installations need many copies of the same program. When a large number of people are using the program, some copies are bound to be accidentally damaged or misplaced, so back up copies are essential. Copies may also be required at remote sites. It is difficult for a company to foresee how many copies of a program it will need, and additional copies may be needed quickly.

Although companies who purchase software in large quantities often get substantial discounts, pirated copies are always cheaper. A manager may want everyone to have a copy, but it may be beyond his department's budget to buy them. In spite of the

fact that companies who participate in piracy face stiff fines, a manager in such a position may very well make as many copies as he needs anyway.

Virtually all companies have official policies forbidding illegal reproduction of copyrighted software, but how well these policies are enforced varies from place to place. Additionally, many employees may be unaware of what is and isn't illegal duplication.

Consequently, it is nearly impossible for firms with large computer installations to prevent all piracy. Software publishers understand this, but they would still like to make as much money as possible on each sale. So, some software publishers decided to sell some company's the right to make a certain number of back-up copies, for the companies use only. In these arrangements, the company has bought the right to make extra back-ups, but not the right to reproduce them for resale or employees personal use. This kind of contract is classified as a licensing agreement rather than a purchase (see Software Licensing in Chapter 9 The Law). It's called site licensing because the right to reproduce the software is usually restricted to one place, the company's location or site, and to copies for official use.

The actual terms of a site licensing contract are worked out individually between publisher and buyer. Publishers who offer site licensing each have their own contracts. A publisher usually will offer the same deal to every company, but is usually willing to change specific details to make a sale.

Microsoft Corp offers a plan that doesn't permit back ups, but gives companies with offenders a easy way out. In this plan, large quantity purchasers receive a discount if the corporation exercises "due diligence in discouraging" illicit copies. If unauthorized copies are found, the company's liability is restricted to the suggested retail price of the software (instead of the usual liability of $50,000 per illegal copy).

Lotus Development Corp and Exxon Corp have worked out a unique site-licensing agreement. Exxon has a version of 1-2-3 running on a mainframe. When a microcomputer user needs the program, it is

downloaded, and the mainframe counts the number of downloads, and then Lotus bills Exxon accordingly.

Site-licensing offers advantages to both software publisher and software buyer. A user, employed at a place with a site licensing agreement, may still be able to remove a copy and give it to his fellow workers, but an arrangement such as this goes a long way toward stopping piracy. The software publisher is still compensated for the extra copies, and the company is relieved of the burden of potentially huge fines and lawsuits. As these contracts become more common, and publishers and users become more comfortable with them, their popularity will continue to grow. Better techniques of keeping track of and paying for copies will be developed. In the future, even fairly small computer bases such as small businesses, schools, and libraries will routinely set up site licensing agreements.

LOGIC BOMBS AND PROGRAM WORMS

One intriguing area of programming relates to programs that change over time by growing, duplicating, or even "exploding", bringing down whole systems. The techniques developed go by names like program worms, logic bombs, program viruses, and self-destructing programs. Each is a little bit different, but all can be very dangerous if misused. These programs do have some legitimate uses. They include:

1. Tracking down portions of old code in a large program, and update it with a revision.

2. Performing diagnostic tests on network systems.

3. Allowing the limited use of a program by having it self destruct after a set number of runs.

4. Performing experiments and simulations by having the program act as a primitive life form.

Program Worms

The name "worm" was first used for a program in a story by John Bruner called Shockwave Rider. It told of an oppressive government that used a huge network of computers to track and control the people. Eventually, a rebel programmer is able to defeat the government by letting loose an unstoppable program named "Tapeworm" which ends up destroying the network.

John Schach and John Hupp, two research programmers at Xerox's Palo Alto Research Center are credited with actually creating the first real program worms in the late 1970's. They were studying the possibility of artificial life by creating programs that would move through computers on a network and replicate themselves on idle computers. The "worm" program could migrate to any accessible computer, then take over its resources for itself. They found it effective for checking network security, and then they created mutations for other functions.

One program worm called "Existential" could stay alive in the network when some machines were turned off, and would display the message "I'M A WORM, CATCH ME IF YOU CAN" on the console of computers it inhabited. Eventually, a worm accidentally mutated, and brought down over 100 computers on the Xerox network. The defective worm had jumped quickly from computer to computer, crashing each one as it went. They spent hours trying to find and destroy worm segments that had gotten into every corner of the research center. Fortunately, it was stopped before it made it's way through a gate which linked the center with other centers all around the country. Later, they developed a special breed of worm called "Killer" which would seek out and destroy other worms on the network.

The most famous worm program was released on unsuspecting Apple users a few years ago. Known as "Killer DOS", it would spread over bulletin boards and on copies of disks by acting like a regular DOS, but actually infecting all the disks and files it

manipulated. Eventually it would strike by formatting disks or scrambling files and directories. Although a fix that would remove the worm was eventually created, it was only after it had spread through entire communities and destroyed a lot of work.

These examples show how dangerous a worm program can be. When a worm program is used maliciously, it is often called a virus program, because of it's ability to multiply by infecting other systems. Although program worms can be used destructively, they can also be used for valuable purposes.

Logic Bombs

A logic Bomb is a program that normally performs a useful function, but upon a special condition will turn and destroy itself, other programs and files, or attempt to bring down the whole computer center or network. A logic bomb can be triggered by running a certain set of data, running a certain number of times, or just hitting a special condition when running.

Logic bombs have been used in the past by disgruntled employees to get revenge, as practical jokes, and by software companies and programmers to collect money owed, or to created limited use programs. As explained in Vol I, a limited use program is one given out as a demo, which will self destruct after certain number of uses. This is effective only if the program can't be copied or if the user doesn't know that it will self destruct. The disk included with this package contains a program that will automatically protect a file in this fashion. It allows the program to run a selected number of times, then destroys itself and displays a message you select. See the disk documentation for further information.

HARDWARE DATA KEYS - A NEW BEGINNING

The ADAPSO Proposal

Vol I of this set of books explains some of the pros and cons of hardware data keys, and recommended that their use be considered by

software publishers. This advice (whether from the first book or from other sources) has been taken very seriously. The Association of Data Processing Service Organizations (ADAPSO) has gone to extraordinary means to propose standards for hardware keys. Although the consensus to use hardware keys is far from unanimous, on October 15th, 1985 ADAPSO published about 100 pages of detailed specifications in their "Proposal for Software Authorization System Standards".

This proposal, which was sent to more than 300 industry leaders, describes a software protection method based on a 3 part system. Part 1 would be in the software, and parts 2 and 3 would be hardware devices. The software lock would be a part of the program designed to communicate with its appropriate hardware key. Without successful communication, it could prevent the program from running, or permit only a portion of the program to run in a limited fashion or in a demonstration mode.

The hardware data key (part 2) would be a small external device that comes with the program, and would communicate with the software lock (part 1) permitting the program to run. It could contain anything from a simple ID number to complex routines that are actually executed inside the key. The key would interface with the computer via the "key ring" (part 3).

The key ring is a device which ADAPSO suggests should be standardized, to hold the users collection of keys for various software products. The key ring contains intelligence, and acts as a traffic manager for communications between the main computer and the keys. Each key ring would have a unique serial number in a masterkey, assigned by a central clearing house set up for that purpose. The key ring would be connected to the computer by an I/O port, and would route information through other devices connected to the same port.

The SAS (Software Authorization System) would communicate with the computer using a complex four level protocol, complete with error checking, a standardized command/response set, false start resistance, communications passthrough, and collision

detection. The key ring would have to be capable of communicating at 19.2K bits per second, have automatic baud rate detection, and contain a 2K buffer.

The Justice Department, Antitrust Division, has issued an initial O.K. to the study of the proposal and has stated that they have no present intention to challenge the plan based on the information that they have received. However, the Microcomputers Management Association has stated it's opposition to the standard on the grounds that it would be too expensive and too time consuming to implement.

ADAPSO is currently in the process of reviewing objections and recommendations to the plan and hopes to gain the approval of industry leaders. The details of the proposal are available to anyone by calling or writing to ADAPSO at:

ADAPSO
1300 North 17th St.
Arlington, VA 22209
(703) 522-5055

Current Activities

With or without a standardized hardware data key plan, some companies are going ahead and implementing their own systems. Most familiar to Atari users is the key used with Paper Clip from Batteries Included. It permits the disk to be copied, but the copies will only work if the hardware key is plugged into the joystick port of the computer. This relatively simple key uses the techniques described in Vol I. The system works. Paper Clip has not been as widely pirated as other programs.

A more complex hardware key has been built by Dallas Semiconductor. Their device, called the 'Electric Key', uses four levels of copy protection, an is powered by a small lithium battery. The key prevents tampering by using an electronic seal, which will destroy the data in the key if it is opened.

Although the key is conceptually the same as the ADAPSO plan, it does not meet their specifications. The key is said to cost approximately $6.50 in quantities of a thousand. It's plugged in

through a computers printer port with an interface device which costs about $18.00 each, in quantities of a thousand. Dallas Semiconductor plans to test market the hardware key on scientific programs for the IBM PC to gauge its acceptance.

Although all this activity in the area of hardware data keys makes their future appear brighter, the cost, complexity, and inconvenience of hardware data keys may still prevent their widespread use in the long run.

MISCELLANEOUS METHODS

Some protection techniques in use have not been easy to classify. These methods are important though, and growing in popularity. This section will discuss a few of the most common.

Random Access Codes and Passwords

Some programs prompt users to enter selected passwords from the documentation before they will run. Each time the program is used, it forces you to enter one of as many as 18 different passwords before it will proceed. Although this method is inconvenient, legitimate users can look up the necessary password with little trouble. Pirates, on the other hand, often can get only incomplete documentation, if any. Programs traded over a modem are especially likely to be lacking documentation.

Infocom has incorporated this technique into one of their adventure games, Spell Breaker. After playing about halfway through the game, the player reaches a door. The door says it needs an answer to a question and the answer is in your guidebook. There are 6 different questions, and they change each time the game is played. The documentation is made to appear trivial, so most pirates don't bother to make a copy. Needless to say, a pirate lacking documentation, who has taken the time to play that far, will be upset. After hitting that door, many pirates have broken down and bought a legitimate copy so they could finish the game.

Partially Functional Copies (Bait & Hook)

A similar concept, equally annoying for pirates, is used in Alternate Reality. The copies seem to run normally at first, and the games seems to play so pirates think their copies are functional. Later, to their dismay, they learn their player is always sickly, and dies soon after he begins playing. The idea is to give the pirates enough of a taste of the program to make them want to buy it.

Elaborate Documentation

Another concept from Infocom is fancy documentation. Some players enjoy the documentation enough to go out a buy a copy of the game, even though they may have a pirate copy at home. One of their newest games, "The Leather Queens of Phobos" includes a beautifully done R-rated color booklet, complete with 3-D glasses and a scratch-and-sniff page. Pirates will have a hard time trying to copy that. Infocom has raised their documentation to a new form of status symbol that is sure to put a dent in piracy.

Support

Another way to differentiate a pirate copy from an original is with support. Many companies have a set policy where no questions will be answered unless the user is a registered purchaser. For a business or productivity program, support can be vital. A company that is responsive to customers also tends to discourage piracy by generating the good will that goes with a well supported product. Let's hope that the trend toward good support continues.

Conclusions

Companies will continue to try to make legitimate copies of their software more desirable than pirated copies. If that trend continues, the piracy problem may begin to disappear.

Chapter 11

A LOOK AHEAD IN SOFTWARE PROTECTION

130XE - NEW POTENTIALS AND PITFALLS

With the introduction of new computers, Atari has opened the door for many new developments. The rise in memory upgrades for the 400/800 and XL series has begun. Now that a standard for increasing memory has been established, the 1 meg XE expansion can't be too far behind.

The Effect on Copies

Now that the 130XE has been released, the trend to develop software upgrades that utilize its extra memory has begun. This causes a headache for pirates, as they can't exchange their disks for new versions. If a pirate wants a new version he must track down an original owner, or buy a copy himself. Widely pirated programs benefit the most from upgrades, because pirates grow to like them, and therefore want the newer, more powerful versions.

Another feature of the new machine will also impact piracy. Flexible operating systems, like that in the 800XL, can be easily modified with software, and the extra memory offers many places to 'hide' extra code for tracking and breaking software. One announced product from Computer Software Services, called The Miracle, will utilize the flexible operating system to monitor and copy software. If it's released and successful, other similar products are sure to follow.

But by far and away the most important impact the 130XE will bring to the industry is the increased support by software developers for Atari computer

systems. Developers see the 130XE as a sign of new life for the whole Atari line. The 130XE will do a lot to insure the continued arrival of new software.

THE FUTURE OF SOFTWARE PROTECTION

With the Atari 8 bit computers over seven years old, the expectation would be that protection technology would be fairly stable and mature. In fact, this area is changing faster than almost any other. New protection and copy methods arise as frequently as new programs. With the new life brought to the 8 bit line by the 130XE, aggressive pricing, new hardware peripherals and unique applications, Atari software, and its protection, is far from stagnant. Although software publishers had taken a strong lead in software protection, new copy utilities and individual efforts have shown a surprising rebound. For the latest news in this area, check your disk included with this package.

Although it hasn't been felt yet in the Atari market, by far the biggest trend in software protection is no protection at all. In the IBM PC market, fully 80% of the business software sold has no protection at all. Although there are many other forces affecting this development, probably the most important force encouraging this trend is the increasing use of hard disks. The high speed and storage capacity of hard disks is lost if the program must be loaded from a floppy disk. Many of the companies releasing unprotected software are spending considerable sums to fund public education to discourage piracy.

Last year has seen the rise of the hardware data key. This trend is only temporary. They are too cumbersome and expensive to become very popular. The ADAPSO policy will fail to gain the widespread acceptance they hope for.

One area that will change rapidly is the law. Sting operations will continue, and computer crime laws will be considerably toughened. Hackers will still make the news occasionally, but as companies become more and more aware of the problems, their

security will get significantly tougher.

The fastest moving area is communications. As modems increase in speed and drop in price, more and more people will join the on-line crowd. What impact this will have on piracy is hard to say. On one hand, it will make it easier meet and trade software, and on the other, modem users often develop a greater sense of community, and even become friends with their favorite software authors.

One trend that discourages piracy is the ever plummeting price of software, particularly older titles. It's not unusual to see original copies of older programs, complete with documentation, for as little as $5.00. New software with a high demand will always demand a premium price, but as more titles grow old, more programs will join the ranks of very inexpensive, good quality software.

The price of blank disks has fallen just as fast. Some people believe that lower blank disk prices have offset the drop in price of legitimate software. Many pirates have claimed that the only reason they copied programs was because the programs were outrageously priced. They claimed that when software prices were reduced to reasonable levels, they would no longer copy programs. Only time will tell if they were speaking the truth.

Some experts feel that public education and awareness is the key to preventing software piracy. To that end, some organizations have begun extensive anti-piracy campaigns.

It seems certain that the problem of piracy will continue to plague software publishers for some time to come. There are no simple solutions. Only when users agree that piracy is not in their best interest will the problem come to an end. Until such a time, the battle between software publishers and pirates will rage on.

SECTION III
THE TOOLS

Disclaimer

The following reviews and opinions are based on extensive study and use of the products described. They represent a detailed look at the usefulness and capabilities of the products, but are not necesarily the last word. Products are updated occasionaly, new features are added, or bugs are fixed. These reviews are based on the newest release of these products (the version number is listed when available) and will be updated upon each new edition of this book. This section attempts to cover the most popular and useful utilities and, needless to say, some may be left out. If your favorite is not here, and you would like to see a review of it included in a future revision of this book, please let us know. Lastly, keep in mind that this is an attempt at an objective and unbiased view of these products, covering the advantages and disadvantages of each.

The reviews all follow the same general format, to make comparisons easier. The format is:

Product Description
- Breif product overview.
- The Hardware
- The Software
- Documentation
- Price

How it Works

Ease of Use
- Installation
- Automatic Copies
- Software Tools
- Support

Net Results
- What it Copies/What Skill is Needed
- Copyable, Useable Copies
- Uses Other Than Copying

Conclusion

You may address all comments, recomendations, etc. directly to Alpha Systems.

Chapter 12
THE HAPPY ENHANCEMENT
AND
— The 1050 Duplicator —

THE HAPPY ENHANCEMENT

Version 7.0

Happy Computers Inc.
PO Box 1268
Morgan Hill, CA 95037
408-779-3830

Product Description

The Happy Enhancement is a disk drive modification and software package desgned to copy protected software and increase the drives operating speed. It is available from Happy Computers, and some mail order houses.

The Hardware

The hardware is available to modify both 810 and 1050 disk drives. In both cases, the hardware consists of a circuit board that is installed in the disk drive. The 810 board contains a custom 4K ROM, two 8K static RAMS, a Quad NAND IC, and Quad OR IC.

The 1050 upgrade board is more like a self contained computer. It contains a custom 4K ROM, three 2K static RAMS, a TTL dual line counter/multiplier, and it's own 6502 microprocessor (the same chip that controls your Atari).

The hardware provides more than just copies; it significantly speeds up your disk drive, while reducing normal wear on the heads. Another extra (only on the 1050 version) is true double density.

The Happy still allows the 1050 to work in single and dual (enhanced) density, but adds the capacity for true double density. These added benefits make The Happy useful even when you are not creating backups.

The Software

The software version 7.0 was released to the public in May, 1986. Although it won't run on many early 810 versions of the Happy Hardware, Happy offers an upgrade chip for $49.95, which makes it compatible with all previous hardware versions.

The software is powerful, but doesn't give a clue to what it is doing. It's difficult to use it for anything besides the built in functions. However, it does have many fine built in functions. It can perform disk diagnostics, make sector copies of unprotected software, copy protected software with several special options, utilize more than 1 Happy drive, and compact protected programs so more than one will fit on a disk. Happy also provides a warp speed DOS, so that other programs can benefit from the fast I/O rates.

The software itself is not protected, but, of course, it is useless without the Happy hardware. Some new features of the most recent release include RAM disk support (130XE or Axalon) for the sector copier option, and compatibility with both 810 and 1050 drives (so that both kinds can work together). Most importantly, this new release contains special Pre-Defined Back-Up (PDB) Files, which specifically back up certain programs which Happy normally cannot copy - but more on that later.

Besides the software included with the Happy package, there is a third party software package, named Happy Archiver Software, which adds much to the usefulness of the Happy Hardware. The Archiver Software, which is not sold by Happy (it is available from other sources) will be fully explained in the next section, on the Archiver/Editor Chip. It allows the Happy hardware to perform all the functions associated with the Archiver. It is extremely useful for studying protection and protecting your own

programs. The Happy Archiver Software usually sells for $39.95 and is copy protected (Happy cannot automatically copy it).

Documentation

The documentation comes in two parts, the installation manual and the usage manual. Both are printed on a dot matrix printer, then Xeroxed and stapled together (a little cheap for a $150 package).

The installation instructions are complete and relatively easy to follow. The software usage section leaves much to be desired. It is heavy on acronyms, like RUT, SCP, HBP, HCP, MDP, PDB, and WSD, which all refer to different parts of the Happy programs. Needless to say, this can make parts of the manual a little cryptic. You can get the basic program functions from the manual, but it contains virtually no technical information on how it works. In fact, they make a point of saying they don't include it because people try to copy their ideas (true in some cases), but it leaves the user to figure out the details on his own.

Price

The list price was recently reduced from $249.95 to $149.95. Some mail order houses offer Happys for a bit less.

How It Works

The Happy Hardware works by first replacing the standard disk drive ROM with its own special 4K version that can perform the extra functions. It also contains a special 4K buffer (6K on the 1050 version) that holds an entire track of data from the disk. That way Happy seems to read data instantly when the track has already been stored in the buffer. Each time a call is issued for a sector, Happy first checks to see if it's stored in the buffer. If it's not, then Happy will read the entire track, so on subsequent calls the data will be in the buffer. The Happy hardware also supports a higher speed transfer rate. When that's coupled with the warp speed software, it can read and write disks

even faster. Finally, the Hardware is programable. Although Happy won't release the details of it's functions, the programability allows them to update the software without replacing the hardware. The exception to that was the 7.0 version for 810 drives; it requires a chip change. It was changed primarily to stop the spread of pirated Happy boards.

Because the Happy Enhancement performs different functions, we'll break them down and look at them one at a time.

Copying

Happy allows the copying of most protected disks in one of two ways. The first relies on the Hardware to read entire tracks of data into the special buffer. As explained in Vol I, a standard Atari drive can read only a single sector at a time, and it relies on the standard ROM and floppy disk controller to the drive to find it. This makes a standard drive very easy to trick. For example, take the case of a duplicate sector. A standard drive looks at the sector header information, and sends the first sector that matches the sector number back. It has no way of knowing if that is the only sector with that number. However, a Happy drive, by reading the whole track, can easily find all the sectors with a matching sector number. Finding the information is only a small part of the copying process. The trick is writing it back out.

The main problem with a standard drive is that it's ability to format is locked in a set pattern by its ROM programming. It can only accept the command to format (and the density desired) and it does the rest. When it's time to write out data, it searches the track for the proper spots, and will only insert the data there. A Happy drive can format the track and write the data any way it wants. In this way, the Happy can copy the format, as well as the data, from the original track. Because that special format is the disks copy protection, Happy can duplicate most disks. This copy method allowed Happy to copy virtually every disk, until 1985, when some new protection methods sprang up.

The new methods of overfilled tracks, short

sectoring, and unstable sectoring (explained in the chapter Disk Specific Protection) either had too much data for a Happy drive to write, or tricked the drive into writing out a track that was different from the original.

Happy's first attempt required the user to slow down the drive to about 269 RPM (288 is normal) to give it the extra time to cram more data onto the track. This method was cumbersome, and still couldn't copy unstable sectors, or an abundance of short sectors. This meant that for a period of 18 months, software publishers took a lead over Happy in protecting their products.

Happy's response finally came in May 1986, in the form of the Pre-Defined Backup (PDB) Files. These files contain the specific patches and/or special track formats needed for many of the most popular programs that used the new protection methods. Below is a list of those files.

Happy 7.0 PDB Files

1) Syncalc type 1 (Synapse)
2) Syncalc type 2 (Synapse)
3) Electronic Arts Programs
4) Synfile
5) Synchron
6) Synstock
7) Alleycat
8) Encounter
9) New York City
10) Electrician
11) Blue Max
12) Quazimodo
13) Dimension X
14) Epyx Games (Kronos Rift, The Eidolon)
15) Questron side 0
16) Questron side 1
17) Questron side 2
18) Questron side 3
19) Microprose, Softee, and Hayden
(Kennedy Approach, Silent Service, XWord Puzzles
V2 #2, Sargon III)
20) Scanalyzer

21) Spy vs Spy
22) Alternate Reality
23) Temple of Asphai
24) Super Bunny
25) Lode Runners Rescue
26) Zorro

To copy one of these programs, you use the Happy Backup Program, and then select the PDB for the specific program you wish to copy. The program makes a backup, then the PDB information is used to patch the parts that are uncopyable. This technique has three problems.

First, not all the backup copies will run on a standard drive. Sometimes some of the data must be moved to alternate tracks, then the Happy Hardware tricks the program into thinking the data came from the correct track. The important part is that some of these backups ONLY run on a Happy Enhanced drive. They will not run on an ordinary drive.

The second problem is more serious. Because the PDB file contains the specific data needed to copy a specific program, Happy's backup method is extremely easy to defeat. All the PDB files have specific sectors and tracks programmed in, so all a publisher has to do to is move the protection by one sector to defeat it. It takes about 10 minutes to change the protection so that the specific PDB information is useless. Also, different versions of a program can have different protection, so again, Happy is defeated. Of course, Happy could always change their PDB files, but, as anyone who has dealt with Happy can tell you, don't hold your breath.

The third problem is that this method won't work on new software or heavily protected software for which there is no PDB file. Basically, it forces you to rely on Happy to send you a new PDB each time a new program comes out. However, Happy states that they will no longer send out information automatically each time a new program is released. It is up to you to contact them, and arrange to get any new PDB files.

Speed

Happy achieves its admirable speed increase in two ways. The first way is the track buffer. This method speeds up all disk usage (except for heavily protected disks, when drive speed must be slowed to normal). It works by reading an entire track into Happy's built in buffer, then sends the sectors to the computer as requested. It still transfers the data at the same speed, but the time it takes to find and read each sector is greatly reduced.

The second speed up method only works when using a Happy Backup Program or one of the Warp Speed DOS's. This method incorporates the track buffer, then it speeds up the data transfer rate to achieve the fastest read/write time from an Atari drive.

Ease Of Use

Installation

The installation in both an 810 and 1050 is relatively easy and straight forward. First, open the drive, then remove the RF shield, remove a few chips, then insert the Happy board. Although quite easy, it will void any warranties on the drive, and can be botched if you're not very careful removing and replacing the chips. It is recommended that someone with experience helps.

Automatic Copies

Automatic copies is where Happy excels. Only a few parameters need to be set before making a backup of any disk that Happy can copy. Most copies can be made with Happy Backup Program, or a PDB. In a few cases, special parameters need to be set, but it's a simple process. However, this ease of use can work against Happy. If the backup won't run, and you've tried changing their few parameters, you're stuck. The Happy will give no indications of what the problem is or how to overcome it. Fortunately, as mentioned earlier, the Happy hardware will run a special version of Archiver Software, which goes a long way toward fixing this.

Software Tools

Happy's other software tools include diagnostics (to check out your drive), compaction options (which sometimes allow more than one backup on a disk), and Warp Speed DOS (explained earlier). All of these options are easy to use once you've done it a few times and know what to expect.

Support

Support is one of Happy's weakest areas. Customer support is very hard to receive, and their attitude, expressed in letters and in the documentation, is 'Don't try'. They tell you not to call if you have a problem backing up a specific program. They offer virtually no technical information except a standard write up. Again, they are very difficult to reach. However, these programs are easy enough to use so that no help should be needed.

Net Results

What it Copies/What Skill Level is Needed.

Happy copies just about everything, and it requires very little skill on the part of the user. There are at least several programs out of those tested that it could not copy, but it does handle all the most popular. The disadvantages are:

1. The only new heavily protected programs it copies are those with specific PDB files. That means it won't work with brand new heavily protected programs.

2. It's simple for software publishers to change the protection so the PDB files no longer work.

3. Some copies require Happy to run.

Copyable, Useable Copies

Although Happy is excellent at making copies, each copy is just as protected, and just as difficult to copy, as the original.

Uses Other Than Copying

These are some of Happy's best features. Besides making copies, it will significantly speeds up the drive. It allows the use of true double density on a 1050 drive, and, in many cases, allows more than one backup per disk.

Conclusion

All in all, Happy, at $149.95, is probably the best buy for a backup program today. It's extra features and ease of use make it a worthwhile investment for any 810 or 1050 disk drive owner who wants to backup software.

THE 1050 DUPLICATOR

Duplication Technologies, Inc.
99 Jericho Tpke, Suite 302A
Jericho, NY 11750

Because of the similarities between this product and the Happy Enhancement, it did not warrant a separate chapter. Below is a summary of the similarities and differences between the two.

As stated, the 1050 Duplicator is very similar to the 1050 Enhancement from Happy Computers. In fact, its hardware features are almost an identical copy of Happy. When the 1050 Duplicator was first released, it sold for much less than the Happy Enhancement, but Happy Computers has dropped its price to match the $149.95 price of this new clone.

The main differences between the two products are listed below.

1. The 1050 Duplicator has no companion 810 version, so the drives will not work together.

2. The software doesn't include an equivalent to Happy's Pre-Defined Backup files, so it doesn't copy newer software.

3. There is no Archiver software available for the Duplicator.

4. The Happy Enhancement has been around for years. Happy Computers is an established company that regularly upgrades their products. Duplication Technologies is a new company whose future survival and support is far from assured. (Duplication Technologies formerly did business as the now defunct Gardner Computing).

Duplication Technologies is promising a future program that will transmit protected programs over a modem. If this product is created, it will be a good plus for their system.

Chapter 13

THE ARCHIVER/EDITOR CHIP

Versions 1.0 - 1.2
Originally From Spartan Software
3417 Nobel Ave N
Crystal, MN 55422
Now available only from distributors and some mail order houses.

Product Description

The Archiver/Editor, also called The Chip, is a backup device available only for 810 drives. A special version of its software is also available for 810 or 1050 drives with Happy Hardware installed. The software with the Archiver is among the most useful that a software publisher or serious student of the art of backups can own. Unfortunately, Spartan Software of Minnesota is now out of business (some of the former members have gone on to form ICD), so availability and support is very limited.

The Hardware

The hardware is a simple 2732 4K EPROM chip which replaces the ROM in an 810 drive. It also requires 3 jumper wires and a few cut traces.

The Software

As stated above, the software is the heart of the Archiver/Editor system. It is broken into several parts.

1. The Archiver - A backup program that allows the setting of several parameters and shows what it's doing as it copies.

2. The Editor - A powerful track and sector editor which allows the visual display of entire tracks, showing duplicates and sector status. It allows sectors and whole tracks to be moved and modified. It also contains such extras as a simple way to change the status of a sector (good for protection or fixing a bad sector), and a disassembler that will show what the sector data would represent in Assembly language.

3. Mapper - Actually maps out the layout of a track on the disk.

4. Formatter - A screen that allows you to set up a track format specifying the order, length, and fill bytes of sectors. It allows you to write a custom format to a disk.

Documentation

The documentation is a spiral (plastic binder) bound book of about 70 pages. It contains not only the instructions on installation and use, but also a section on the theory of disk format, and a section of useful hints to help you get more out of it. It's somewhat outdated, and doesn't talk about some new protection techniques, but this is to be expected, as the book was printed a few years ago.

Price

The original price varied according to the distributor, but usually sold for $129.95. The software for Happy drives lists at $39.95. Because the Archiver is a simple EPROM, it was widely pirated and sold (of course, with no support from Spartan) for prices between $15.00 and $80.00. Recently the original has been discounted to about $75.00.

How it Works

The Archiver works by reprograming the disk drive to respond to additional functions. The Archiver chip replaces the drive's standard ROM, and it contains the programing needed for tasks such as reading and writing entire tracks. The Archiver acts like a normal drive, until it receives a special OPEN password, which activates its unique features.

Ease of Use

Installation

Installation of the Archiver Hardware is tricky and should only be attempted by an experienced solderer. It requires the dismantling of the drive and the replacement of a chip. The most difficult part involves cutting 3 circuits (traces) on the drive controller board and soldering in 3 new connections with jumper wires. The work is delicate and must be done neatly and carefully. Of course it voids any warranty on the drive, but 810s are out of warranty by now anyway. Although it is easier to install than some of the new memory upgrades, the task should not be taken lightly.

Automatic Copies

The Archiver is no longer very good at automatic copies. In early 1984, it could copy just about everything, but as time went by, it fell further and further behind. The newer releases of the software, versions 1.1 and 1.2 helped, but only temporarily. The software allows the setting of several parameters which sometimes helps, but the Archivers strength is not in it's ability to make automatic copies.

Software Tools

The software tools are the Archivers strong point. In fact, with proper knowledge, skill, and determination, the Archiver software will let you backup as much, if not more, than other systems. The drawback is that the skill level required to do that is very high. It is not as time consuming as

breaking a program by hand, but it does require careful study, as well as trial and error. The real power of Archiver lies in its ability to show what the disk contains, and thus makes it easier to understand what the protection is doing.

The outstanding feature of the Archiver software is its ability to protect software. Whether it's software given to friends, or publishers protecting commercial programs, the Archiver lets you lay down the protection exactly the way you want it. Of course, your software must still check for the protection, but that information is explained in this book series.

NOTE: For non-programers, the disk in our package (included with your purchase) contains a program that will allow you to automatically make your files look for any protection you want. With the Archiver/Editor and the software on this disk you can publish and protect your software with ease.

Support

Obviously support is a problem, since the producers are out of business. Limited support may be available from the place where it is purchased. Upgrades will probably not be offered, unless this product is picked up by another company.

Net Results

What it Copies/What Skill Level is Needed

The Archivers copying ability depends on the skill level of the user. It has enough features so that it can assist an advanced user at copying almost anything, but on its own, it mainly copies older, less protected, titles.

Copyable, Useable Copies

The Archiver's copies are like Happy's copies, usually as protected as the original. Archiver is more of a help if you are trying to remove the protection and make unprotected copies.

Uses Other Than Copying

The Archiver's only other use is for studying protection and applying your own protection to disks. At this, it performs well.

CONCLUSION

The Archiver/Editor is a useful and well designed product. It is somewhat difficult to install, and the hardware is only available for 810 disk drives. If you already own a 810 or 1050 Happy Enhancement, then the Archiver software at $39.95 is a worthwhile investment.

Chapter 14

THE IMPOSSIBLE

Computer Software Services
PO Box 17660
Rochester, NY 14617

Product Description

The Impossible takes a new approach to disk back ups. Instead of modifying the disk drive with hardware, the computer is modified. It has its own advantages and disadvantages. It is available for the 400, 800 (with ROM revision B), and the 800XL.

Hardware

The hardware for the 800 version consists of a small circuit board with its own ROM program, 4K of RAM, and various NAND connectors. The advertisements describe it as a 4K static RAM pack, which leads many people to believe it is a cartridge. It is not a cartridge; it's actually a device which is attached to the operating system board by removing three chips and replacing them with this board, and the chips wired to it. They tell you, correctly, that no disk drive modifications are needed, but modifications to your computer are needed. Also, you will have wires connected to a switch hanging out of your computer. The switch turns the Impossible on and off, and allows you to open up the extra 4K of memory if desired (no commercial programs except theirs uses the extra memory).

Switching the switch is required to run the software.

The 800XL version is very similar to the 800 version, except that it's harder to install. More on this in the section on installation.

Software

The Impossible software performs one function; it allows you to back up certain programs. There are virtually no extras in the way of disk analysis, extra speed, etc. The software merely loads and offers the options of a BASIC or Assembler program in a normal or H-P (High Performance) mode. It is claimed that the H-P mode can copy certain programs that the normal mode can't, but it does require more memory, and therefore can't copy as large of a program.

Documentation

The documentation is several pages of Xeroxed paper, stapled together in the corner. It seems a little cheap for a $149.95 package, but it's quite complete. Because its functions are limited, there's not a whole lot that needs to be said.

How it Works

The Impossible is unique in the way it makes backups, and therefore offers advantages and disadvantages as compared to the other methods. Their special hardware actually changes the operating system of the Atari so that once activated, all calls to the disk drive are channeled through their program. Their program sits in the unused 4K of address space at BFFF to D000. It's called by their special hardware 'hook' in the operating system whenever disk I/O is performed. This means that the Impossible will work with any drive, which is a big plus for people with Indus, Ranas, etc.

The backups are made by switching their switch to the Impossible position (which activates their software) and loading their disk. Their disk installs the backup software which prepares it to monitor while you load the program to copy. Next, you

insert the disk to back up, and hit START. It loads in its normal fashion, but it's now being monitored by the hidden software.

The program loads, checks its protection, and runs normally (since the original disk is in the drive). Once it is entirely done loading from the disk, you push OPTION and SELECT at the same time, and the Impossible program takes over again. Now it knows all the sectors that your program read, and it rereads each one to store its data and status. Finally, it prompts you to insert your destination disk, so it can write out all this information to a normally formatted disk.

Running the Copies

Running the copies is simple. Insert the backup disk in the drive, move the switch to the Impossible position, and turn the computer on. The program will seem to load and run like the original. What's really happening is that each time it calls for a disk sector, the Impossible software intercepts it and goes to the disk itself to get the data and status that it needs. In this way, the Impossible tricks the program into thinking that the original disk with the original program is in the drive, when it's actually a normally formatted disk with a very different layout from the original!

This method of operation leads to some big advantages and disadvantages over other systems. The advantages are:

1. It can work with any disk drive (as long as it can read the program) regardless of brand.

2. The backups are normally formatted boot disks that can be copied at will.

3. It reduces wear on the drive, since the drive doesn't have to read protection or strange formats.

It's disadvantages, listed below, can be a problem, though:

1. The backup you make will ONLY RUN ON YOUR SYSTEM. Because the Impossible is required to run the backups, they are useless to anyone without an Impossible. At $149.95, it can be expensive to equip all your computers with an Impossible.

NOTE: The producer, Computer Software Services, has just released a program called the XL Mate, which allows some of these backups to run on a non-Impossible 800XL system, after the XL Mate software is run. Unfortunately, the XL Mate software is heavily protected, so to run the backups on another system a copy of this software must be purchased. In addition, it only works with some of the programs the Impossible can backup.

2. Because this program doesn't really copy the protection, but only tricks the computer into running the program, it's easy to defeat. A software writer looking to prevent the Impossible from working can merely have his program look for it, and if found, lock up the system. CSS does deserve some credit for disguising this device. It is difficult to detect without some experimentation. Although most Software companies can easily defeat the Impossible, surprisingly many have not. Apparently, the Impossible is not seen as much of a threat because of its restrictions.

Ease of Use

Installation

Both the 400/800 and 800XL installations are more difficult than would be expected from their advertisements. In fact, the 800XL models (the ones with soldered in chips) require an extremely good technician for installation. It should not be attempted by an amateur hobbyist. Fortunately, CSS offers free installation, the customer pays only for the shipping costs both ways.

If you own a 400, 800, or an early 800XL

(without soldered chips) you can probably install it yourself. To install it, you must open the computer, remove a few chips, and replace them with some others. If you're careful, and don't bend any pins, it should go without a hitch.

Automatic Copies

The Impossible, in general, is easy to use. In some special cases it requires some thought and patience to use. For example, to copy a graphic adventure, you must make two copies, one with the Impossible,and one with a sector copier. When the program is running, you must switch disks after the protection is checked.

Support

The support from CSS is good. They seem responsive to questions and try to help.

Net Results

The Impossible can copy a good percentage of all available software. It's a well implemented, original idea. Most copies are simple to make and easy to use. The chief drawback is that 800 owners require the Impossible to run their backups. One plus is that older 800 can have an extra 4K of memory.

Conclusion

At $149.95, the Impossible makes a good backup alternative for those who don't have Atari drives. Although software companies can stop the Impossible from working, most have not done so. It can easily back up the majority of software for the Atari.

Chapter 15

THE SCANALYZER

Alpha Systems
4435 Maplepark Rd
Stow, Ohio 44224

Product Description

NOTE: The Scanalyzer is an Alpha Systems product. Every attempt has been made to keep this review as fair and as unbiased as possible.

The Scanalyzer is different from the other products reviewed here. It is a software only product, it needs no additional hardware, or hardware modifications. Rather than recreating protection schemes on backup disks, Scanalyzer provides utilities that can be used to remove the protection altogether, resulting in a completely unprotected program. To produce such a copy does, however, require more programming skill and knowledge than the other products reviewed.

The Software

The Scanalyzer is a program analyzer package that consists of several utility modules. Together with time and patience, they provide a skilled programmer with all the things needed to break protected programs by hand.

The software modules are BASIC Lister, Directory Finder, Cartridge Reader, Disk Scanner,

Data Analyzer, and Disk Back-up.

The BASIC lister module will list any BASIC program, including unlistable ones. It will also restore the variable table, if that is necessary, and remove the protection from BASIC programs. The listing can be sent to the printer or the screen, or it can be saved to a disk file. In addition to creating backups of unlistable BASIC programs, this module is an excellent tool for recovering partially damaged BASIC files.

The Directory Finder module will search an entire disk for directories, hidden and normal, pausing as it displays each one. This module will also remove a file from a disk with a hidden directory, and transfer it to a normally formatted disk.

The Disk Scanner scans the disk and identifies and displays the protection that it encounters. It has two main parts. The first is a 'Fast Scan', which scans the disks and identifies protected sectors and the form of protection used. The second part is an 'Analyze and Edit' mode that displays the sector data in both hex and ATASCII format. In this mode, you can print the data on a printer, change the data, write the sector to a disk, or scan the sector for protection or a duplicate sector. It also traces sector links, provides a map of VTOC usage, and gives a detailed directory showing starting sectors, etc.

The Cartridge Reader module does just that - it will read a 4, 8, or 16K cartridge, and save the cartridge program as a binary load file. The file can then be used with the Data Analyzer to remove the protection. Or, it can be used with the Impersonator package. Some early cartridge programs like Chess, Basketball, etc. have no protection, and these cartridges can be backed up to disk simply by running this module. One drawback with this module is that the current version (3.3) requires that XL or XE owners use the Atari Translator disk before running it. The rest of the Scanalyzer does not need the Translator at all.

The Data Analyzer module converts pure data into readable assembly language. It displays it on the

screen, and can also save it to a file or dump it to a printer. The resulting file can be modified and debugged with the Atari Assembler Editor cartridge. It can accept code from RAM, a binary load disk file, disk boot sectors, any arbitrary sectors, or any DOS non-binary file. It inserts the standard Atari labels, as well as a couple of special labels (to help identify protection).

The Disk Backup has several parts. A normal sector copier, a 130XE one pass sector copier, and two techniques of creating bad or unreadable sectors. The ability to write bad sectors does not help backup much software any more, and is not very useful.

Documentation

The documentation is a 30 page booklet. It contains concise, easy to follow instructions on how to use each module of the program.

Price

List price for the Scanalyzer is $29.95.

How it Works

Most backup systems focus on ways to recreate the original protection, or to deceive the program into believing that the correct protection is intact, whether or not it actually is. The result is a copy that is just as protected as the original, or one that requires special hardware and software to run properly.

Scanalyzer takes a completely different approach to making backup copies. It provides a programmer with the necessary tools to find and remove the protection from a program. The result is a completely unprotected copy, that can be reproduced or modified at will. The catch is that it takes a fairly high degree of skill to achieve those results.

A person wishing to make backups would first scan the disk to find and identify the protection, then remove the code that checks for the protection, and make any other modifications. Persons who are less skilled can learn a lot by using

this program. It allows you to study the methods used by professional programers, as well as learn about disk and cartridge protection. The BASIC Lister is great for adding that one feature missing from a program, or to study professionally written BASIC programs.

Ease of Use

Since there's no hardware, there's nothing to install. Scanalyzer will make automatic sector copies, but if the program is protected the backup will not run. It's good for making automatic copies of BASIC programs. Using the bad sector writer requires a little more work, but not much software is protected with plain bad sectors anymore, so it's unlikely that anyone will need to use this option anyway. It's easy to make copies of cartridge files, but again, if they are protected they will need work before they can be run (or they can be used with the Impersonator).

Software Tools

This is the area where Scanalyzer's value shows. The BASIC Lister is excellent for recovering damaged BASIC files. The data analyzer is great for someone who is learning Assembly language, because it gives you the ability to view program code that has been written by professionals. It is also an excellent tool for someone who wants to study software protection.

Support

Alpha Systems will provide assistance to any registered user who has a problem. Letters with technical questions get a personal reply. The operators at the order line cannot answer any questions, but the customer service staff at the customer service line will answer any questions they can. If it's a difficult problem, and they can't solve it right away, they will arrange for you to talk to someone who can.

Net Results

What it Copies/What Skill is Needed

Scanalyzer will list protected BASIC programs, and backup programs that can be copied with a sector copier. Scanalyzer provides the tools to back up virtually anything, but the skill level needed to do that is very high. The harder the protection, the more skill, knowledge, and effort it requires.

Copyable, Useable Copies

Copyable, Useable copies is where Scanalyzer excels. If the user has the skill, the copies are completely copyable, useable and modifyable. They can be converted to binary load files, and easily transmitted over modems.

Uses Other Than Copying

Again, this is one of Scanalyzer strong points. Any BASIC programmer will appreciate the ability to rescue a partially damaged program. The other tools are useful for both experienced programmers who wish to customize their software, and beginning Assembly language programmers to study Assembly language programs.

Conclusion

For a person who wants to break software by hand and produce completely unprotected versions of protected software, Scanalyzer is one of the best programs available. However, it's not for people who simply want to stick in a disk, press a button, and produce another protected backup, or for people who have no interest in programming. Because the program modules are full of versatile, useful features, Scanalyzer is a worthwhile investment for programmers and students of software protection.

Chapter 16

THE PILL, THE SUPER PILL AND THE IMPERSONATOR

The Pill ($69.95) and the Super Pill ($79.95)
Computer Software Services
PO Box 17660
Rochester, NY 14617

The Impersonator ($29.95)
Alpha Systems
4435 Maplepark Rd
Stow, Ohio 44224

Product Description

Each of these devices fall into the same category of cartridge backup systems. They each have their own pluses and minuses, but all work in the same way. They are discussed together for the sake of comparison.

Hardware

Each package contains a special cartridge used for running cartridge backups. The Impersonator is a simple circuit board, the Pill is a cartridge with a toggle switch, and the Super Pill is a cartridge in a plastic case. They all serve the same purpose, but each works a little differently.

Software

All three packages give you a cartridge saving program and a menu program to run cartridges that

have been saved to disk. The Impersonator has an additional program which lets it work together with the Scanalyzer to modify cartridge programs.

Documentation

All three packages are well documented, and the documentation explains their functions well. Because of the limited scope of these products, not much explanation is needed.

Price

The Pill lists for $69.95, the Super Pill for $79.95, and the Impersonator for $29.95. They all do the same thing, but each works a little differently.

How it Works

These devices all work by the principles explained in the chapter on cartridge protection (see Pseudo Cartridges). Basically, a special program runs which prompts you to insert the cartridge you want to copy into the computer. Next the cartridge data is saved to a disk file. In the case of the Pill and the Super Pill, the data file will run only with their special menu program. The Impersonator saves the program as a regular binary load file.

Running the programs can be a little tricky. For the Pill, first the Pill cartridge, with the switch off, is inserted. The Super Pill is just inserted. The Impersonator goes in later.

Next, the disk file which contains the cartridge data is loaded. Special built-in menus are used with the Pill and Super Pill. The Impersonator can be loaded with its own menu, DOS option L, or any other menu program. Once the file is loaded, the Super Pill begins to run the file immediately. The Pill stops, prompts you to turn on the switch, and then the cartridge program runs. The Impersonator stops, prompts you to insert the Impersonator cartridge, then begins to run.

In all cases, the program loads into RAM, the writes to RAM are disabled by the cartridges, then the program runs. The Super Pill is switched automatically by the program. The Pill is switched manually. The Impersonator is just inserted.

Ease of Use

Making backups copies is the same on all systems. Running the backups is easiest with the Super Pill, because the switching is automatic. The Pill and The Impersonator require you to change a switch or insert the cartridge at the appropriate time. All of these require the special cartridge to run the backup programs. The Impersonator does permit backups of unprotected cartridges to run without the Impersonator cartridge.

Once the backups are made, all of them are easy to use, but making the backups can sometimes be difficult. All the systems require you to insert the cartridge to be copied with the computer turned on. This can lead to a computer lock up, so the right touch must be used. With a little practice anyone can make and run backups with ease.

Net Results

All three devices copy all Atari computer cartridges with a few exceptions. None can copy bank select cartridges. The only bank select cartridges available at this time are Bounty Bob from Big Five, and most of the language cartridges from OSS. Only the Impersonator can copy old cartridges that used the right hand cartridge slots on the 800 computers, Monkey Wrench is the only commercial example.

All copies saved to disk can be copied again at will, but most won't run without the special backup cartridge installed. Only the Impersonator allows the files to be mixed with other kinds of files on the same disk. The Super Pill allows backups to run the easiest.

Conclusion

With the limited number of cartridges being released for the Atari, these systems only make sense for someone with a large cartridge collection.

They also make sense for someone who travels, as the disk files save space. Of the three systems reviewed, each has its own good and bad points. The Impersonator, at $29.95, is the cheapest, but the Super Pill, at $79.95, is the easiest to use. It is up to the individual to decide which one fills his needs and is best.

GLOSSARY

Access Levels - This determines what a user can see and do on a BBS. Higher access levels place fewer restrictions on user activities, lower levels have more restrictions.

Account - A password/ID combination that permits a user to perform specific functions on an on-line system.

ANI Numbers - Special telephone numbers that, when called, will identify the number of the telephone that the call is being placed from.

Back Doors - Special accounts used by manufacturers, repair personnel, programmers, etc, to access a computer system. They are usually known only to the person who set them up.

Bad Data Marks - Marks used to identify the type of data in a disk sector. Data marks other than $FB are bad. They are used in copy protection.

Bad Sectors - Missing sectors or sectors on a disk which contain unreadable data.

Bank Select Cartridges - Cartridges that can switch between two or more separate banks of memory. They are used for cartridges with more than 16K.

Baud Rate - The rate at which data is tranmitted over a communications channel.

BBS - Bulletin Board Sytem. See boards.

Beta Version - a test version of unreleased software. Beta versions are ususally not completely debugged.

Boards - Electronic Bulletin Board Systems. They can be large, public systems, such as Comp-U-Serve, or small and private, like many pirate boards. They allow other computer users to call by modem to exchange data and information.

Boot - The process of loading a program from a disk or tape into the computer.

Boot Disk - A disk with programs that will load automatically when the computer is turned on.

Boot Sectors - The sectors on a disk where the loading information is stored. Sectors 1, 2, and 3 are set aside for this purpose.

Boxes, Black Boxes - Boxes are hardware devices used by phreakers to control or decieve the phone company's computers. Black boxes are the most widely used.

Common Channel Interoffice Switching - A system used by the telephone comany, with one line for voice, and a separate line for control signals.

Compiler - a piece of software which converts program code into machine language.

Copyright - The right to make and distribute copies of a work. Copyrights protect only the expression of an idea, not the idea itself.

CRC Errors - Errors that occur when the CRC bytes do not match the data on a disk sector.

Cracking - the practice of breaking into computer systems, often using telecommunications.

Custom Format - A layout of sectors and tracks on a disk that does not match the standard disk format.

Data Encryption - See Encryption.

Data Key - see Hardware Data Key.

Detokenizer - A program that converts BASIC tokens into the BASIC commands they represent. It can be used to LIST 'unLISTable' BASIC programs.

Directory Analysis - The process of analyzing a disk's directory.

Disassembler - a program which will convert machine language into Assembly Language for eay analysis and use.

DOS - Disk Operating System. It controls the operations of the disk drive.

Duplicate Sectors - Two sectors with the same number, but each contains different data. They are used in software protection.

Electronic Switching System (ESS) - A system permitting the telephone company to trace calls in a matter of seconds.

Encryption - The process of converting data into a code through the use of a block of data called the key.

EPROM - Eraseable Programable Read Only Memory. Memory chips that can be programmed and erased with ultraviolet light.

EPROM Burner - A hardware device that can read and write to an EPROM or PROM chip.

Format - The layout of data on a disk or program tape. Standard disk format is 40 tracks with 18 sectors per track, and 128 bytes of data in each sector.

Front Operation - A facade used to cover up a BBS devoted to piracy.

Fuzzy Sectors - See Unstable secotrs.

Hackers - Dedicated computer hobbyists who enjoy the technical side of computing.

Hacking - Using a trial and error process of working out codes or numbers, such as MCI access codes. Also refers to quickly programming or changing programs.

Hardware Data Key - A hardware device used to protect a program. The software may be copyable, but the key must be plugged into the computer in order for the software to run properly.

Key Ring - a proposed device which would hold several different hardware data keys, and permit the apropriate key to communicate with the computer.

Licensing - The practice of selling only a copy of a program and the right to use it, and forbidding unauthorized duplication and distribution of the software.

Load Analysis - Observing and analyzing a programs loading process.

Logic Bombs - A program that will work under normal circumstances, but when triggered, will change it's function in a possibly destructive fashion.

Loops - Special circuits used to test phone lines.

Modem - A device which permits two or more computers to exchange information over telephone lines.

On-Line Systems - Computer systems which can be accessed through telecommunications.

verfilled Tracks - Tracks with more than 18 sectors.

Partial Directory - a directory containing incomplete file information.

Password - a sequence of characters which permits a user access to a system. If the password is entered incorrectly, the user is denied access.

Patent - The right to exclusively create and distribute an item. A patent protects both the expression and the idea.

Phantom Sectors - See Unstable sectors.

Phreaks, Phone Phreaks - Persons whose hobby is working with telecommunications and the telephone system. They are usually involved in illegal phone use.

Pirate - Person who makes and/or distributes illegal copies of copyrighted programs.

Pirate Boards - Bulletin Boards with the primary purpose of posting and exchanging pirated software and information on copying programs.

Profitier - A person who pirates software for profit.

Program Worms (Program Viruses) - Programs that can duplicate themselves, migrate between systems on a network, and utilize idle computer time for their own purposes.

Protocol - A standard procedure used when transmitting data that enables the sender to properly encode the information, and the receiver to properly decipher it.

Pseudo Cartridges - Cartridges used with cartridge backup systems to trick the computer into believing that an actual cartridge program is installed.

Pseudo Directory - A directory which contains false or inaccurate information about the disk files.

Reverse Engineering - a method of duplication. It's done by studying the original and its construction, and creating a duplicate the same way the original was built.

Sector Analysis - The study of the sectors on a disk, including determining statuses, and examining the format of the disk.

Self Destructing Programs - Programs which will destroy themselves under a specific set of conditions. See also logic bombs.

Short Sectors - Sectors which contain less than 128 bytes fo data.

Site Licensing - The practice of selling a number of copies of software and the right to use them. The arrangement usually includes limited liablility for illegal copies, and/or the right to make a limited number of copies for company use only (not for commercial distribution).

Software Licensing - See Licensing.

SYSOP - System Operator. The owner or person in charge of a bulletin board.

Token - A number which represents a BASIC command, and is used to save storage space.

Trade Secret - A recipe or process that makes a product unique. It must be kept confidential.

Trojan Horse Programs - Programs, often destructive, with deceptive, innocent sounding names or functions.

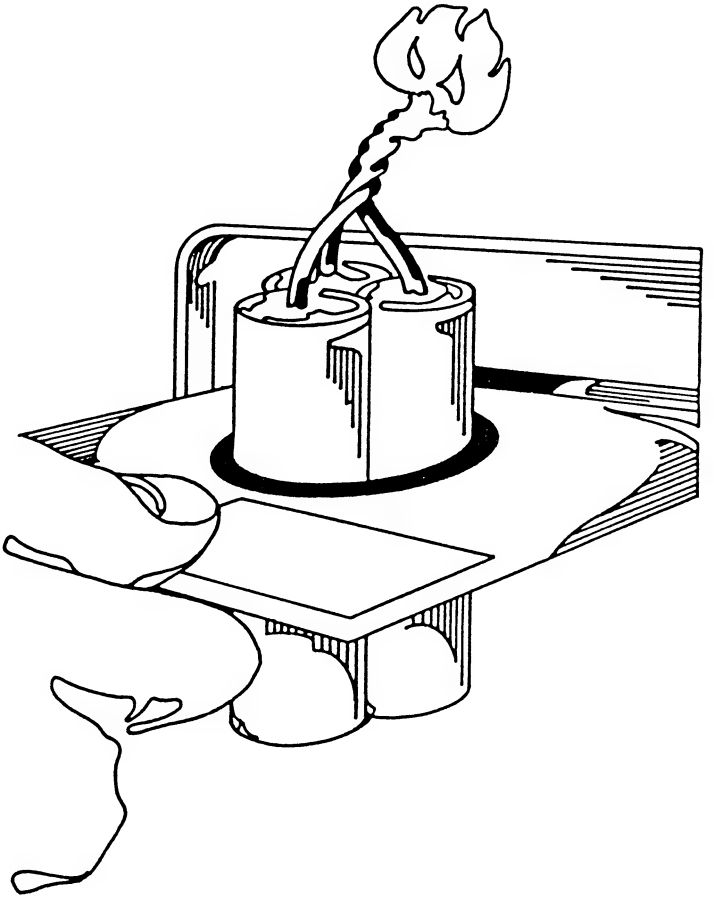
2600 Magazine - A magazine devoted to hackers and phreakers.

Uniform Commercial Code - the body of law which governs most business transactions inside the United States.

Unstable Sectors - Sectors which contain data which changes every time it is read.

VTOC - Volume Table Of Contents. It keeps track of which disk sectors are full or free.

SECTION IV
ADVANCED PROTECTION TECHNIQUES
DISK DOCUMENTATION



ADVANCED PROTECTION TECHNIQUES DISK UTILITIES

By George J Polly
George Morrison
Al wylcznski

INTRODUCTION

With the growth of piracy in recent years, even the novice programmer needs a quick and easy way to protect his programs. Too often programmers spend weeks on protecting their software. ADVANCED PROTECTION TECHNIQUES DISK UTILITIES does all of this in a matter of minutes. It can add a password, encrypt, check for bad sectors, or limit the number of times a binary file will run. It can modify any sector on the disk, and easily mark the sectors to be used for protection. It will also encrypt data files, so you can insure that your data is kept private.

Before you start, please read the accompanying book, and be familiar with passwords, encryption, bad sectors, etc. Those concepts are important to understanding how these programs work.

To help insure the safety of programs and data protected with this package, Alpha Systems won't release any technical information about the internal workings of these programs or it's protection. Therefore, it is extremely important to make and keep working backups of your source files. Once you have applied your protection, your program will be as inaccessible to you, as it will be to anyone else.

THE DISK FILES

The ADVANCED PROTECTION TECHNIQUES DISK UTILITIES disk contains several files. The main programs are called PRO, PRO2, and DATA. Those files are run automatically (by the AUTORUN.SYS) when the disk is booted. They make up the DISK UTILITIES described below. The DISK UTILITIES disk also contains a file called NEWS.TXT. This is a text file that contains the latest news in the software protection field. It can be displayed and printed

from the DISK UTILITIES menu. The disk also contains a file called PUBLIC. This is a public domain binary load file. It will come in handy for trying out the protection methods that are created by the DISK UTILITIES. Just copy it to a blank disk, and try your hand at protecting it using the DISK UTILITIES options. The disk also contains a series of specially protected sectors for you to work with. Those sectors will be explained later.

LOADING THE DISK UTILITIES

1. Insert the disk in drive 1.
2. Hold down the OPTION button on XL and XE computers (remove the BASIC cartridge from others), and turn the computer on.
3. After loading, ADVANCED PROTECTION TECHNIQUES DISK UTILITIES will ask for the default drive. This is the drive which will be used if no other drive is specified. It can be changed from the ANALYZE AND EDIT screen. For now, just press 'I' and 'RETURN' to go to the main menu.

MAIN MENU

The main menu consists of four options:

DISK EDITOR
PROTECT BINARY FILE
DE/ENCRYPT FILE
LATEST NEWS IN SOFTWARE PROTECTION

The first option, DISK EDITOR, is used to modify the data on a disk and to scan for bad and duplicate sectors. The second option, PROTECT BINARY FILE, will protect binary load files by any number of methods. The next option, DE/ENCRYPT FILE is used to make any file unreadable and unuseable, then restore it at a later time. Finally, the LATEST NEWS option will display the latest news in software protection that has occurred since the book was printed.

To select one of these options use the '=' key

move the highlighted block down, and the '-' key to move it up. When the desired option is highlighted, press the 'RETURN' key and that option will run. These instructions apply to any menu of this kind.

THE DISK EDITOR

There are five options in the DISK EDITOR

ANALYZE AND EDIT
SCAN FOR PROTECTION
DISPLAY THE DIRECTORY
MODIFY VTOC
RETURN TO MAIN MENU

The first option, ANALYZE AND EDIT, will display the contents of any sector on the disk and allow you to edit the information that it contains.

First, you will be asked to enter the sector number to start with. Type any sector number from 1 to 720, and hit 'RETURN'. If you have a 1050 disk drive, and are looking at an enhanced density disk, you may enter any sector number from 1 to 1040. Next, you will see the sector data displayed in both hex and ATASCII format. Several options will be displayed to the bottom of the screen. The HELP option is the one you will probably want to use first. Press 'H' and 'RETURN', and the program will display a help screen which describes the many other functions available in this mode.

The second option, SCAN FOR PROTECTION, allows you to scan any number of sectors for protection. First, a prompt for the starting sector will appear. Type in the number of the starting sector and press the 'RETURN' key. Do the same for the ending sector, and the scan will begin. It will print the sector number, and indicate whether it is good, bad, or duplicate, and displays the status (if it was bad). It will continue until the ending sector is reached. Refer to the accompanying book if you need more information about the specific protection techniques and how they are used.

The third option, DISPLAY DIRECTORY, will

display the directory of the disk in the default drive. The directory includes the starting sector and length of each file. It also shows any deleted files, and marks them for easy identification.

The fourth option, MODIFY VTOC displays the VTOC and allows it to be modified. This function supports one of the most important features of a protected disk. It allows you to mark sectors that are being used so that DOS won't write over them.

Suppose you wanted to make a disk where bad sectors are scattered throughout. The problem is how to copy your programs to the disk avoiding the bad sectors, but using the sectors near them. This feature makes it simple to mark off the sectors that you want to protect before copying your files to the disk.

After selecting this option, the screen will display the VTOC of the disk currently in the disk drive. Across the top is the sector number, and down the side is the track number (NOTE: two track are displayed per horizontal line). A 'Y' on the table indicates the sector is used, a 'N' indicates the sector is free. At the top of the screen, the number of free sectors is displayed in parenthesis, and next to it is the number of the sector which is being modified. To change the value from 'Y' to 'N', or vice versa, press the 'SPACE' bar. To move to another sector, use the arrow and control keys. When you've finished, press the 'W' key to write out the modified VTOC. If you don't want to save it, press the 'RETURN' key to exit this option. Once your VTOC is changed, DOS will automatically skip those marked sectors when copying files to the disk. This greatly simplifies disk protection.

PROTECT BINARY FILE

The 'PROTECT BINARY FILE' option will add protection to any ordinary Atari DOS 2.5 or 2.0 binary load file. That means it works with almost any program that can be loaded with DOS option 'L' (try the file called 'PUBLIC' on your disk). It can

add a password, encrypt, limit use, check sector status, or any combination of the four. This is done by changing the program in your binary file. The binary file is loaded exactly as before, and it will operate the same, except that it will check for the protection you specify. NOTE: This feature will not work on files which load into memory locations $600 to $700 (1536 to 1792). Only one limited use program can be put on each disk.

After you have chosen this option, a list of the 4 protection options will appear followed by the word 'OFF'. To use one of the protection methods on your file, use the '=' and '-' keys to select the options you want, then press 'RETURN' to turn them ON. Any combination (or all) the methods can be used on each file. After you have selected all the options you want, select FINISHED. If you no longer want to protect a binary file, select MAIN MENU.

If the PASSWORD option is chosen, the program will ask for password, up to 8 characters long. You can use any numbers, letters, or graphics characters in your password. Do not forget this password! When you run the protected binary file, it will ask for the password, and it will not continue until the correct password is entered.

NOTE: Once again, we will remind you that once a file is protected, it is not easily undone, so be sure to keep an unprotected backup. If you forget your password, and do not have an unprotected copy, we will not be able to help you.

The ENCRYPTION option will encrypt a binary file so it can not be disassembled or modified, but it will still run. Once it has been turned 'ON', the ENCRYPTION option will run automatically. This feature is especially good for protecting your name and copyright information in your programs. Often, pirates will find and change them using a sector editor (such as the one contained in this program). If you select the ENCRYPTION option to protect your file, it will be almost impossible for anyone to alter it.

The LIMIT USAGE option will permit the binary

file to run only a set number of times. After this option is chosen, enter the number of times the binary file is to run. Then, enter a message to be displayed when the limit is reached, such as 'SORRY CHARLIE'. Just press 'RETURN' for no message. After the useage limit has been reached, and the file has displayed your message, it can do one of two things to insure that it cannot be used again. The binary file can destroy itself, or the file can format the entire disk, destroying itself as well as any other files that happen to be there. The DESTROY FILE option actually writes over the file, so no 'UNDELETE' programs will work.
NOTE: The FORMAT DISK option should not be used unless the entire disk is being protected. If the protected file can be moved to another disk, and the FORMAT DISK option is used, it will destroy itself, and any other files on the disk. A powerful feature such as this must be used with caution and maturity. Stick to the DESTROY FILE option if you will be protecting only one file.

Backups are even more critical for limited use programs. Once the program has run your selected number of times, it will destroy itself any cannot be recovered.

The final option, SECTOR CHECK, will check the status of up to four sectors. First, enter the status the sector should be, and then the sector number itself. The possible statuses are: 1 for a bad sector, 2 for bad sector with some data, and 3 for duplicate or unstable sectors. If you wish to check less then 4 sectors, enter the status 4 after completing the desired number of sectors.

These options cause the protected file to automatically check the protection when they run. In this case, option 1 will check for any kind of 'bad' sector at the location you specify. You must put a bad sector at that location in order for the program to run. Option 2 says you must have a 'bad' sector which contains good data that you specify. Only CRC errors and bad data marks will qualify here. Option 3 checks for duplicate or unstable sectors, in

other words, sectors which seem to change each time you read them. Obviously, if you want to use these options to protect your programs, you must be able to create these kinds of custom formats. Two methods of writing bad sectors are described in Vol I, Atari Software Protection Techniques. Creating the other kinds of protection usually requires special hardware, like those described in the Reviews section of this book.

Finally, decide what should be done if the correct status is not found. The DESTROY FILE and FORMAT DISK options are the same as those used in the Limited Use method. The LOCK UP option will cause the computer to lock up if the protection is not found. This is the preferred method here, since occasionally, even the original disk can fail to pass the protection check.

Finally, you will be asked if everything is correct. If it is, type a 'Y' for yes, and the program will continue on. If you type 'N' for no, the program will return to the main menu.

When the program has prepared the protection for your file, it will ask for the name of the binary file to protect. Enter the filename. If the file is in a drive other than the default drive, you must enter "D: and the appropriate drive number before the filename. Press the 'RETURN' key without entering anything to display all the files on the disk in the default drive. After you have entered the filename, press 'RETURN'. The program will find the file, and ask for the name of the output file, which will be the protected binary file. Press the 'RETURN' key to use the same filename, and the unprotected file will be replaced by the protected file. The program will then protect the binary file.

DECRYPT/ENCRYPT FILE

The DECRYPT/ENCRYPT FILE option allows you to encrypt any file type according to your own password, and decrypt it at a later date. Remember your password, because the file can never be

decrypted without the correct password. By encrypting the file, you change each byte to other (seemingly random) bytes and the file cannot be used. Only by decrypting it with the proper key can it be restored. This is good for any kind of data that must be kept confidential. Things like your charge accounts, Comp-U-Serve ID numbers, and long distances access codes are best stored on your computer in encrypted form.

After this option is selected, choose either ENCRYPT a file or DECRYPT a file, then enter a password of up to 20 characters. Next, enter the name of the file to encrypt, or press 'RETURN' to display a directory of the disk. Finally, enter the output filename, or press 'RETURN' to use the same filename.

DISPLAY LATEST NEWS

Because of the nature of the ever changing world of software protection, some important events can occur after each edition of the book goes to press. These events are recorded on the disk between updates to the book, so you can be sure that you are getting the most up to date information available. The DISPLAY LATEST NEWS option will display or print the latest protection news which was not printed in the book. After choosing this option, you can either display the file on the screen, or print it on your printer.

PROTECTION DEMOS

The disk included in your package also contains a number of the most advanced protection methods available. These are included as examples for you to study and learn from. Below is the table of protected sectors, and the protection used on each.

SECTOR	PROTECTION METHOD
690	Standard 'bad' or 'missing' sector
692	A duplicate sector - Two separate sectors with the same number
695	A short sector (10 bytes long)
700	A CRC error, contains data of all Xs.
704	A bad data mark, contains data of all Ys.
709	An unstable sector - seems to fill in with random data on different reads

Another good use of those sectors is for protecting your own files. An easy way is to use 'HAPPY' (or another backup device) to copy track 38 from this disk to your own disk. Track 38 includes the protected sectors 690, 692, 695, and 700. You can use the utilities to make your program check for any or all of these sectors before running. Remember to use the MODIFY VTOC option of the DISK EDITOR to mark these sectors as 'used'. Otherwise, you may accidentally write over them. Also, it is recommended that only one of these sectors should be checked. Checking them all could slow the loading process down considerably.

OTHER ALPHA SYSTEMS PRODUCTS

Atari Software Protection Techniques - Volume I of the Protection Techniques series, also written by George Morrison. It includes sections on: Protection of BASIC Programs, Cassette Protection, Hiding Directories & VTOCs, Bad & Misassigned Sectors, ROM Protection & Copy Techniques, Hardware Data Keys, Legal Protection, and Coercive Protection Techniques. A must for anyone who owns Advanced Atari Protection Techniques.

Scanalyzer - A #1 bestseller. For details, see the review in Chapter 15.

Impersonator - Another Top Ten package, reviewed here in Chapter 16.

Magniprint II+ - Reviewers have said "Magniprint II+ is by far the BEST graphics screen dump program available...Nothing else comes even close." Prints graphics 9 in 16 shades of grey. Prints graphics 8 and 7.5 pictures in your choice of grey shades. Prints 6ft posters you have to see to belive!

PARROT - The ultimate sound digitizer. Record anything, voice, music, airplane engines, in true digital form. Playback the sound through your TV speaker, without additional hardware. Incorporate the sounds into your own BASIC programs. Manipulate them any way you like. It even turns your Atari keyboard into a unique musical instrument.

Write or call for a full catalog of our other fine products.

LIMITED WARRANTY

Alpha Systems warrants the original purchaser of this computer software product that the recording medium on which the software programs are recorded will be free from defects in materials and workmanship for ninety days from the date of purchase. Defective media returned by the purchaser during that ninety day period will be replaced without charge, provided that the returned media have not been subjected to misuse, damage, or excessive wear.

Following the initial ninety day warranty period, defective media will be replaced for a replacement fee of $6.50.

Defective media should be returned to:

ALPHA SYSTEMS
4435 Maplepark RD
Stow, Ohio, 44224

in protective packaging accompanied by: (1) a brief statement describing the defect; (2) a $6.50 check or money order (if beyond the ninety day warranty period); (3) your return address; (4) the problem disk.

What is Not Covered by this Warranty

This warranty does not apply to the software programs themselves. the programs are provided "as is".

This warranty is in lieu of all other warranties, whether oral or written, express or implied. Any implied warranties, including imputed warranties of merchantability and fitness for a particular purpose, are limited in duration to ninety days from the date of purchase. Alpha Systems shall not be liable for incidental or consequential damage for breach of any express or implied warranty.

The provisions of the foregoing warranty are subject to the laws of the state in which the disk is purchased. Such laws may broaden the warranty protection available to the purchaser of the disk.

Tell Us What You Think

We at Alpha Systems are sincerely interested in bringing you the best possible products at the lowest possible prices. Please write us if you experience any difficulties with our products, or have any comments or ideas for improvements. We will do our best to make our products better meet your needs. When you write, please enclose the following: 1) Your name, address, and phone number. 2) Your comments, or a description of your problem. 3) A description of your system. 4) If you are reporting a problem, please also include a description of what you were doing when the problem occurred, any printouts or other output showing the problem if possible, and any suggestions you may have regarding the cause and solution.

The frustration of ruined software has touched every computer owner. Damaged software can wipe out days of work or leave you unable to complete a critical job. The only reasonable insurance against such losses is to keep back-up copies.

But the critical need for back-ups is overshadowed by today's unprecedented software piracy. The duplication and exchange of copyrighted software products has cost software publishers over 800 million dollars this year alone.

Today's pirates employ increasingly resourceful means of copying and distributing copyrighted software. Pirated programs posted on electronic bulletin boards can be transmitted (often using illegal access codes on long distance carriers) around the globe.

Today's software publishers are employing increasingly sophisticated and unusual techniques to combat this problem.

This guidebook and the accompanying disk programs will reveal for the first time the state of the art of software protection methods and the techniques used to overcome them. It covers, in complete detail, the most complex protection schemes available today. This book covers the technical details of piracy as well as the social changes and motivations that encourage piracy. It discusses the technical aspects of the protection methods most likely to appear in the future, and provides a clear explanation of where these trends in software protection are leading.